



MEMOIRE

Présenté par

FRIKI Yasser

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Preference Elicitation in Social Recommender Systems
'L'éllicitation des préférences dans les systèmes de recommandation sociaux'

Soutenu le : 25/06/ 2023

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Présidente	Mme. MAKHLOUF Amina	MCB	Chadli Bendjedid El-Tarf
Rapporteur	Mme. MAATALLAH Majda	MCB	Chadli Bendjedid El-Tarf
Examineur	Mr. BETOUIL Ali Abdelatif	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2022/2023

Remerciements

Je tiens à exprimer ma profonde gratitude envers tous ceux qui ont contribué à ma formation et à la réalisation de ma thèse. Tout d'abord, je remercie chaleureusement mon encadrant, *Mme. MAATALLAH* pour son soutien constant, ses conseils éclairés et son expertise précieuse tout au long de ce parcours.

Je souhaite également adresser mes remerciements les plus sincères à tous mes professeurs qui ont contribué à ma formation. Leur savoir, leur passion pour l'enseignement et leur dévouement ont été une source d'inspiration et ont grandement enrichi mon parcours académique.

Un sincère remerciement également à tous les membres du jury d'avoir accepté l'évaluation de mon travail avec attention et objectivité.

J'adresse mes remerciements à mes collègues de recherche pour leur collaboration et leur aide précieuse durant cette aventure académique.

Enfin, je suis reconnaissant(e) envers ma famille pour leur amour, leur soutien inconditionnel et leur encouragement tout au long de cette étape importante de ma vie.

Merci du fond du cœur à toutes ces personnes qui ont contribué à ma réussite académique.

À mes *parents*, mes *sœurs*, mes chers camarades *Ammar*, *Zakí*, *Hythème*, ainsi qu'à tous mes collègues,

Je tiens à dédier ces mots empreints de gratitude et d'affection. Votre présence et votre soutien ont été des piliers essentiels tout au long de mon parcours académique.

À mes *parents*, vous avez été mes premiers mentors, mes guides et mes plus grands soutiens. Votre amour inconditionnel, vos encouragements constants et votre confiance en moi m'ont donné la force nécessaire pour surmonter les défis et atteindre mes objectifs.

À mes *sœurs*, vous avez été mes complices, mes confidentes et mes inspirations. Votre soutien indéfectible et votre présence bienveillante ont été un moteur dans ma quête de réussite.

À mes chers camarades *Ammar*, *Zakí*, *Hythème*, et à tous mes collègues, nous avons partagé des moments d'apprentissage, de travail acharné et de rires. Votre collaboration, vos discussions stimulantes et votre solidarité m'ont aidé à grandir académiquement et personnellement.

Ensemble, vous avez formé une équipe solidaire qui m'a soutenu tout au long de mon cheminement. Vos encouragements, et votre confiance ont été les moteurs qui m'ont poussé à donner le meilleur de moi-même.

Du fond du cœur, je vous remercie pour votre amour, votre soutien et votre amitié indéfectibles. Cette réussite est également la vôtre.

Avec une gratitude infinie.

Yasser

Ce mémoire se focalise sur l'élicitation des préférences dans les Systèmes de Recommandation Sociaux en utilisant des techniques du Traitement Automatique du Langage Naturel. L'étude inclut une analyse approfondie du domaine de la recommandation sociale et de l'élicitation des préférences, en mettant l'accent sur l'analyse des sentiments. Pour effectuer cette dernière, on a utilisé le modèle **AraBERT** récemment développé par Google, qui améliore le traitement des requêtes, la prédiction et la génération de texte en Arabe. Notre objectif est d'exploiter les avantages **d'AraBERT** pour l'élicitation des préférences à partir des commentaires en arabe et l'incorporation de ces préférences dans le processus de la recommandation afin d'améliorer les performances des systèmes de recommandations sociaux en termes de pertinence.

Un système de recommandation social des restaurants nommé '**ArbPrefRec**' basé sur les préférences extraites depuis les commentaires des utilisateurs arabophones a été implémenté avec le langage de programmation python avec une interface conviviale.

Une série de test a été effectuée afin de démontrer la performance du système. En comparons leur performance avec les modèles **LSTM** et **CNN**. Les résultats trouvés en étaient satisfaisants du point de vue de la classification des sentiments et les recommandations générées.

Mots clés : système de recommandation social, élicitation des préférences, analyse des sentiments, recommandation hybrides, AraBert.

This thesis focuses on preference elicitation in Social Recommendation Systems using Natural Language Processing techniques. The study includes an in-depth analysis of the domain of social recommendation and preference elicitation, with a focus on sentiment analysis. To carry out the latter, we used the **AraBERT** model recently developed by Google, which improves query processing, prediction and text generation in Arabic. Our aim is to exploit the advantages of **AraBERT** for eliciting preferences from Arabic comments and incorporating these preferences into the recommendation process in order to improve the performance of social recommendation systems in terms of relevance.

A social restaurant recommendation system called '**ArbPrefRec**' based on preferences extracted from Arabic user comments was implemented using the Python programming language with a user-friendly interface.

A series of tests was carried out to demonstrate the performance of the system. We compared its performance with that of the **LSTM** and **CNN** models. The results were satisfactory in terms of the classification of sentiments and the recommendations generated.

Keywords : social recommendation system, preference elicitation, sentiment analysis, hybrid recommendation, AraBert.

تركز هذه الأطروحة على استنباط الأفضلية في أنظمة التوصيات الاجتماعية باستخدام تقنيات معالجة اللغة الطبيعية. تتضمن الدراسة تحليلاً متعمقاً لمجال التوصية الاجتماعية واستئارة الأفضليات، مع التركيز على تحليل المشاعر. لتنفيذ هذا الأخير، استخدمنا نموذج **AraBERT** الذي طوره Google مؤخراً، والذي يحسن معالجة الاستفسارات والتنبؤ وتوليد النصوص باللغة العربية. هدفنا هو استغلال مزايا **AraBERT** للحصول على أفضليات من التعليقات العربية وإدراج هذه الأفضليات في عملية التوصية من أجل تحسين أداء أنظمة التوصيات الاجتماعية من حيث الأهمية.

تم تنفيذ نظام توصيات المطاعم الاجتماعية المسمى **'ArbPrefRec'** بناءً على التفضيلات المستخلصة من تعليقات المستخدم العربي باستخدام لغة برمجة Python مع واجهة سهلة الاستخدام.

وأجريت سلسلة من الاختبارات لإثبات أداء النظام. قارنا أدائها بأداء طرازي **LSTM** و **CNN** وكانت النتائج مرضية من حيث تصنيف المشاعر والتوصيات الصادرة.

الكلمات الرئيسية: نظام التوصية الاجتماعية، استنباط الأفضليات، تحليل المشاعر، التوصية الهجينة، AraBert.

Table des matières

Remerciements.....	i
Dédicace.....	ii
Résumé.....	iii
Abstract.....	iv
ملخص.....	v
Table des matières.....	vi
Liste des figures.....	x
Liste des tableaux.....	xii
Liste des acronymes.....	xiii
Introduction Générale.....	1
Chapitre 1 : Etat de l'Art.....	4
1.1. Introduction.....	4
1.2. Les systèmes de recommandations sociaux :.....	4
1.2.1. Définitions.....	5
1.2.2. Les Types Des Systèmes De Recommandation Sociaux :.....	6
a. SRS Basés Sur Le Contexte (Context-Aware) :.....	6
b. SRS Basés Sur Les Mots-Clés (Tag) :.....	7
c. SRS Basés Sur La Confiance (Trust) :.....	7
d. SRS Basés Sur Les Données Temporelles (Temporal Data) :.....	7
e. SRS Basés Sur Les Inter-Domains (Cross Domain) :.....	8
1.2.3. Les Méthodes utilisées dans Les SRS :.....	9
a. SRS Basés Sur L'apprentissage Profond (Deep Learning-Based) :.....	9
b. SRS Basés Sur Le Filtrage Collaboratif (Collaborative Filtering-Based) :.....	9
c. SRS Hybrides (Hybrid Techniques) :.....	10
d. SRS Basés Sur Les Ensembles Flous (Fuzzy-Based) :.....	10
e. SRS Basés Sur Le Clustering (Clustering-Based):.....	10
f. Comparaison des méthodes :.....	11

1.2.4.	Les Défis Et Les Problèmes :	12
a.	La Diversité :	12
b.	Le Problème De Démarrage À Froid (Cold Start) :	12
c.	La rareté des données (Sparsity) :	12
d.	La Confiance :	13
e.	L'Élicitation Des Préférences :	13
1.2.5.	Travaux Existants sur Les SRS :	13
1.3.	L'Élicitation Des Préférences Dans Les SRS:	15
1.3.1.	Introduction :	15
1.3.2.	Définition :	15
1.3.3.	Les méthodes utilisées dans l'élicitation des préférences :	16
a.	Méthodes Basées Sur Des Questionnaires :	16
b.	Méthodes Bayésiennes :	16
c.	Méthodes Basées Sur l'analyse des sentiments (Sentiment Analysis) :	17
1.3.4.	Les Difficultés et les Problèmes de l'Élicitation des Préférences dans Les SRS :	17
a.	L'Absence De Feedback Explicite :	17
b.	Le problème De Démarrage À Froid :	18
c.	Le Problème De La Confidentialité :	18
d.	Le Problème De La Qualité Des Données :	18
e.	Le problème De la Diversité :	19
1.4.	L'élicitation des préférences à partir des commentaires :	19
1.5.	Travaux existants sur l'élicitation des préférences à partir des commentaires arabes :	20
1.6.	Le modèle AraBERT :	21
1.6.1.	Les entrées du modèle :	22
1.6.2.	Tokenisations :	22
1.6.3.	Les procédures de pré-entraînement et le Réglage fin de BERT :	25
a.	Pré-entraînement (Pre-training) :	25
b.	Réglage fin (Fine-tuning) :	26
1.7.	Conclusion :	26
Chapitre 2 : Conception et implémentation.....		28

2.1.	Introduction.....	28
2.2.	Contexte.....	28
2.3.	Problématique.....	28
2.4.	Solution Proposée :.....	29
2.5.	Motivation et objectifs.....	30
2.6.	Conception du système « ArbPrefRec » :.....	30
2.6.1.	ETAPE 1 : L'Elicitation Des Préférences :.....	31
a.	Préparation Des Données :.....	32
b.	Configuration Du Modèle Arabert :.....	35
c.	Apprentissage et Evaluation Du Modèle :.....	37
2.6.2.	ETAPE 2 : Génération Des Recommandations basées Préférences :.....	40
a.	La Recommandation collaborative :.....	41
b.	La Recommandation basée sur le contenu :.....	43
c.	La Recommandation hybride basée sentiments :.....	44
2.7.	Conclusion :.....	46
Chapitre 3 : Expérimentations et tests.....		47
3.1.	Introduction.....	47
3.2.	Base de données (DataSet) :.....	47
3.3.	Outils de développements :.....	48
3.3.1.	Langage de programmation (python) :.....	48
3.3.2.	Les bibliothèques utilisées :.....	48
3.3.3.	Le matériel utilisé :.....	49
3.4.	Les métriques d'évaluation :.....	50
3.4.	Evaluation du système ArbPrefRec et comparaison des résultats :.....	50
3.4.1.	Évaluation du modèle d'analyse de sentiment :.....	50
a.	Performance du système pour 3 époques :.....	51
b.	Performance du système pour 10 époques :.....	52
c.	Performance du système dans 20 époques :.....	53
3.4.2.	Comparaison des résultats :.....	56
3.5.	Evaluation de la recommandation hybride basée préférences:.....	59

3.5.1. Réglage des paramètres :.....	60
3.5.2. Évaluation des performances du système de recommandation ArbPrefRec :.....	61
3.5.3. Comparaison des résultats de recommandation :.....	62
3.6. Présentation de l'interface du système de recommandation ArbPrefRec :.....	64
3.7. Conclusion.....	67
Conclusion générale et Perspectives	68
Références	70
A. Références Bibliographiques.....	70
B. Références Web (Techniques).....	75
Annexe A.....	76
Annexe B.....	77

Liste des figures

Figure 1.1. Schéma représentatif des relations entre les utilisateurs et les objets[4]	5
Figure 1.2. Les Relations user-user et user-items[4].....	6
Figure 1.3. l'architecture de AraBERT.[48]	21
Figure 1.4. L'entrée du modèle arabert. [W2].....	22
Figure 1.5. Résultat de pré-traitement de texte	22
Figure 1.6. La différence entre la segmentation de texte avec (Wordpiece et Farasa)	23
Figure 1.7. L'Architecture du model Transformer. [W2]	24
Figure 1.8. Versions de model Arabert (BASE et LARGE).[W2].....	25
Figure 1.9. Les procédures de pré-entraînement et de Réglage fin de Arabert. [W2]	25
Figure 1.10. Architecture du réglage fin (Fine-tuning). [W2].....	26
Figure 2.1. Organigramme représentatif des processus de réalisation de système ArbPrefRec. ...	31
Figure 2.2. Les étapes de l'éllicitation des préférences dans le système ArbPrefRec.....	31
Figure 2.3. Chargement et affichage des premières lignes du Dataset.....	32
Figure 2.4. Le jeu de données après le nettoyage et le prétraitement.....	33
Figure 2.5. Division des données en ensembles d'entraînement et de test.	33
Figure 2.6. Affectation des données au dictionnaire DatasetDict.	34
Figure 2.7. le contenu de dictionnaire après la tokenisation.	35
Figure 2.8. La configuration du modèle utilisé dans le système ArbPrefRec.	36
Figure 2.9. Chargeur de données utilisé.....	37
Figure 2.10. Réglage des paramètres d'entraînement.	37
Figure 2.11. Évolution des performances du système.....	38
Figure 2.12. Performance d'entraînement du système ArbPerfRec.....	40
Figure 2.13. Les étapes de la recommandation hybride.....	41
Figure 2.14. Matrice utilisateur-item.....	42
Figure 2.15. Matrice de la similarité user-to-user.....	42
Figure 2.16. Le classement des items basé sur les sentiments	43
Figure 2.17. Le classement des items basé sur les commentaires.....	44
Figure 2.18. La Recommandation hybride.....	45
Figure 3.1. Performance du système en 3 époques.	51
Figure 3.2. Perte Vs. Précision du système en 3 époques.	51
Figure 3.3. Performance du système en 10 époques.....	52
Figure 3.4. Précision Vs. Perte en 10 époques.....	53

Figure 3.5. Performance du système en 20 époques.	54
Figure 3.6. Précision Vs. Perte en 20 époques.	54
Figure 3.7. La Courbe ROC et matrice de confusion.	55
Figure 3.8. Performance du modèle LSTM.	57
Figure 3.9. Performance du modèle CNN.	58
Figure 3.10. Comparaison entre les trois modèles.	58
Figure 3.11. l'effet des valeurs de paramètre sur le nombre de recommandation.	60
Figure 3.12. Taux de pertinence pour différents nombres d'utilisateurs.	61
Figure 3.13. Comparaison des résultats de recommandation pour TOP-10.	63
Figure 3.14. Comparaison des résultats de recommandation pour Top-20.	63
Figure 3.15. L'interface de système ArbPrefRec.	65
Figure 3.16. Affichage des résultats de recommandations.	65
Figure 3.17. Résultats de prédiction de système ArbPrefRec sur les commentaires.	66

Liste des tableaux

Tableau 1. Les avantages et inconvénients des méthodes citées.....	11
Tableau 2. Tableau récapitulatif des travaux sur les SRS	14
Tableau 3. Tableau récapitulatif des travaux sur l'élicitation des préférences à partir des commentaires arabes	20
Tableau 4. La configuration des paramètres utilisés.....	51
Tableau 5. Comparaison entre les trois modèles sur la partie de test.....	59

Liste des acronymes

SR	Système de recommandation
SRS	Système de recommandation sociale
CBF	Filtrage basé contenu (<i>Content-based-filtrring</i>)
CF	Filtrage collaboratif (<i>colaborativefiltering</i>)
NLP	Traitement automatique de langage naturel (<i>Natural languageprocecing</i>)
SA	Analyse des sentiments (<i>sentiment analysis</i>)
MLM	Modélisation du langage masqué (<i>MaskedLanguage Modeling</i>)
CLS	Classification
QA	Réponse aux questions (<i>Question answering</i>)
BERT	<i>Bidirectional Encoder Representationsfrom Transformers</i>
CSV	Valeurs séparées par des virgules (<i>Comma-Separated Values</i>)
LSTM	Long short termmemory
CNN	Réseaux de neurone Convolutionnel (<i>Convolutif Network Neuronal</i>)
ROC	Caractéristique d'exploitation du récepteur (<i>Receiver Operating Characteristic</i>)

Introduction Générale

Dans le monde numérisé de nos jours, les utilisateurs sont confrontés à la difficulté de naviguer efficacement et de trouver les bonnes ressources en raison de la richesse du contenu et de la quantité immense de l'information disponible en ligne. Afin de répondre à ce défi, les Systèmes de Recommandations (SR) ont été apparus pour aider les utilisateurs à trouver des recommandations personnalisées qui répondent à leurs goûts et préférences.

L'utilisation des systèmes de recommandation est devenue indispensable afin de suggérer des contenus adéquats aux préférences des utilisateurs dans la majorité des sites et applications web, comme il est le cas dans les réseaux sociaux. Ces derniers sont devenus l'outil le plus utilisé au monde par toutes les catégories de personnes, où les utilisateurs jouent le rôle de consommateurs et de fournisseurs d'information en même temps. De plus, Ils produisent quotidiennement une énorme quantité de données et peuvent fournir leurs avis sur les produits et les publications disponibles sous forme de commentaires ou d'évaluations implicites (j'aime/j'aime pas), ce qui augmente considérablement la quantité d'information et du contenu disponible et présente un défi supplémentaire pour extraire des informations de préférences pertinentes et générer des recommandations précises.

En raison de cet excès d'information, les Systèmes de Recommandation Sociaux (SRS) visent à filtrer et à présenter des suggestions personnalisées aux utilisateurs en fonction de leurs préférences. Où, l'élicitation des préférences constitue le processus d'identification et de compréhension des préférences des utilisateurs à partir de leurs commentaires et évaluations fournies.

Récemment, les utilisateurs arabophones des réseaux sociaux favorisaient l'utilisation de la langue arabe pour fournir leurs commentaires, et elle est devenue une langue émergente sur les réseaux sociaux. Cependant, le traitement des commentaires rédigés en arabe pose des défis uniques en ce qui concerne l'élicitation des préférences des utilisateurs, en raison des caractéristiques linguistiques spécifiques et des nuances culturelles et dialectales de la langue arabe. Ce qui rend plus difficile l'interprétation des commentaires avec précision et qui complique d'avantage le processus de compréhension des préférences des utilisateurs. Ce qui nous a conduit à poser la problématique suivante :

Comment peut-on extraire efficacement les préférences des utilisateurs à partir de leurs commentaires saisis en Arabe afin de leur suggérer des recommandations pertinentes et personnalisées dans les réseaux sociaux ?

Afin de répondre à la problématique posée, ce mémoire présente un système de recommandation social qui permet le traitement des avis des utilisateurs arabophones, à propos d'un produit, sous forme textuelle. Ensuite, une analyse des sentiments sera effectuée pour déduire le sentiment correspondant à chaque commentaire donné, soit positif ou négatif. Enfin, une nouvelle approche de recommandation basée sur les sentiments et les préférences des utilisateurs sera présentée afin de la personnaliser et d'augmenter la performance du système.

L'élicitation des préférences à partir des commentaires nécessite l'utilisation des outils du Traitement du Langage Naturel (Natural Language Processing NLP) qui est un sous-domaine de l'apprentissage automatique, visant à donner aux programmes informatiques la capacité de comprendre et de traiter le langage humain. Ces dernières années, l'introduction de BERT[1] (**Bidirectional Encoder Representations from Transformers**) par Google AI a révolutionné le domaine NLP et a démontré des performances exceptionnelles dans ses diverses tâches et a repoussé les limites de la compréhension et de l'analyse du langage. Une version du langage BERT pour le traitement de la langue Arabe a vu le jour, récemment, appelée AraBert [2].

L'objectif principal de ce travail est d'explorer et de relever les défis de l'élicitation des préférences à partir des commentaires écrits en arabes dans les SRS, en s'appuyant sur des techniques avancées de traitement du langage naturel (NLP), en particulier le modèle Arabert[2], afin d'extraire et d'analyser efficacement les préférences des utilisateurs.

Donc, cette recherche vise à améliorer la précision et l'efficacité des SRS, en donnant aux utilisateurs des suggestions plus satisfaisantes, en acquérant des connaissances plus approfondies sur les préférences des utilisateurs depuis leurs commentaires.

Ce mémoire est organisé en trois chapitres comme suit :

- Dans le premier chapitre, nous présentons un état de l'art sur les Systèmes de Recommandations Sociaux, à travers une illustration et présentation des différentes approches, méthodes et techniques de l'élicitation des préférences utilisées. De plus, les principaux concepts du modèle araBERT de Google sont présentés afin de comprendre son fonctionnement.
- Le deuxième chapitre présente la partie conception du système proposé, en commençant par l'architecture du modèle et une explication détaillée de l'algorithme utilisé. Ensuite, le processus de la nouvelle recommandation hybride basée sur les préférences et sentiments est expliqué.

- Le troisième chapitre, présente l'implémentation de notre système proposé et les différentes expériences menées avec résultats obtenus et leurs analyses, en plus des comparaisons avec d'autres modèles.
- A la fin, nous terminerons par une conclusion générale et quelques perspectives.

1.1. Introduction

L'intelligence artificielle a connu un très grand succès ces dernières années, et son utilisation est devenue indispensable dans plusieurs domaines. L'un de ces domaines est les systèmes de recommandation (SR), qui ont vu le jour en raison de la croissance exponentielle des données générées dans le monde entier. De façon générale le but principal de ces systèmes est d'orienter les utilisateurs vers le contenu qui leur intéressent beaucoup, comme Netflix et Amazon.

L'idée de base des systèmes de recommandation (SR) est d'utiliser les diverses sources de données pour déduire les intérêts des clients. L'entité à laquelle la recommandation est fournie est appelée **l'utilisateur** et le produit recommandé est appelé un **article ou item**. Par conséquent, l'analyse des recommandations est souvent fondée sur l'interaction antérieure entre les utilisateurs et les items, car les intérêts et les tendances passés sont souvent de bons indicateurs des choix futurs.

À cause de la grande utilisation de la technologie dans notre vie quotidienne, on ne peut pas nier son contrôle et son influence sur nos actions, surtout que le monde passe la plupart de son temps sur les réseaux sociaux. Par conséquent, les systèmes de recommandations doivent être développés et améliorés pour pouvoir fournir et proposer du contenu personnalisé aux utilisateurs des réseaux sociaux.

Donc, les réseaux sociaux utilisent les SR pour personnaliser le contenu aux utilisateurs et ainsi améliorer l'expérience des utilisateurs. D'autre part, les réseaux sociaux améliorent la qualité de la télédétection en recommandant des éléments pertinents aux utilisateurs en fonction de leurs préférences, de leurs relations et de leurs expériences dans les réseaux sociaux.

Dans ce qui suit, nous présentons en détails les systèmes de recommandation sociaux, ainsi que l'élicitation des préférences avec les défis et les travaux existants dans l'état de l'art.

1.2. Les systèmes de recommandations sociaux :

Les systèmes de recommandations sociaux (SRS) occupent une partie importante des systèmes de recommandations, mais jusqu'à maintenant il n'y a pas une définition littérale.

Les sites de médias sociaux sont devenus extrêmement populaires ces dernières années, les sites de partage de photos et de vidéos comme : Flickr et YouTube, les systèmes de blogues et de

wiki comme : Blogger et Wikipedia, les sites de marquage social tel que: Delicious, les sites de réseaux sociaux comme MySpace et Facebook, et les sites de micro-blogue comme Twitter. Des millions d'utilisateurs sont actifs quotidiennement dans ces sites, créant de l'information riche en ligne qui n'a pas été disponible avant.[3]

1.2.1. Définitions

❖ **Définition générale :**

La recommandation sociale où les SRS sont définis comme tout système de recommandations qui cible les domaines des médias sociaux. [4]

❖ **Définition étroite :**

La recommandation sociale est toute recommandation avec les relations sociales en ligne comme un intrant supplémentaire, c'est-à-dire : augmenter un moteur de recommandation existant avec des données sociales supplémentaires. Les relations sociales peuvent être des relations de confiance, d'amitié et d'adhésion.[4]

Dans un SRS, les utilisateurs sont connectés entre eux via un réseau, et leurs interactions, telles que les J'aime, les partages et les commentaires, sont utilisées pour produire des recommandations. Le système utilise des algorithmes afin d'analyser les connexions sociales entre les utilisateurs et la manière dont ils interagissent avec le contenu.

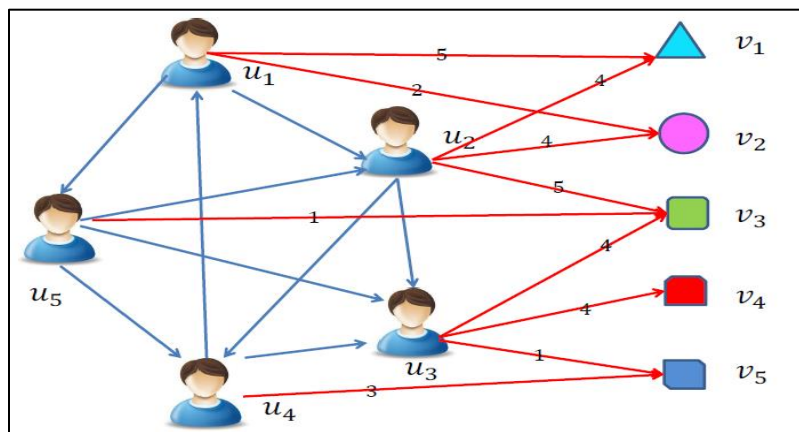


Figure 1.1. Schéma représentatif des relations entre les utilisateurs et les objets[4]

‘**Social Recommender System**’ est l’intégration des médias sociaux et **RS**. Les réseaux sociaux et **RS** partagent leurs avantages mutuels. L’immense expansion des réseaux sociaux a créé des occasions pour les chercheurs d’analyser les réseaux sociaux et d’utiliser leurs résultats dans les **RS**. [5]

Par exemple, si un utilisateur a plusieurs amis qui aiment un film en particulier, le SRS peut le recommander à l'utilisateur selon les préférences de ses amis. Les recommandations

généérées par un SRS sont généralement plus personnalisées et pertinentes que celles générées par les systèmes de recommandation traditionnels, qui ne prennent en considération que les préférences et les comportements individuels des utilisateurs. [5]

Les systèmes de recommandations classiques sont divisés généralement en deux grands types : les SR basées sur le contenu et les SR basées sur le filtrage collaboratif, il existe aussi les SR hybrides, ce type combine entre les techniques des deux types précédents.

Les systèmes de recommandations basées sur le **contenu** se concentrent sur l'historique des évaluations de l'utilisateur actuel et sur la similarité entre les éléments afin de proposer des recommandations, tandis que les systèmes de recommandation **collaborative** utilisent les évaluations implicites ou explicites d'autres utilisateurs et les comparaisons de l'historique des évaluations de l'utilisateur actuel avec d'autres utilisateurs pour fournir des recommandations significatives à l'utilisateur actuel. [6]

Par contre dans les systèmes de SRS, des aspects et des composants des recommandateurs traditionnels sont explicitement conçus en utilisant des entités sociales et des contextes sociaux. Une variante importante des systèmes de recommandation sociale (filtrage social) est obtenue en remplaçant le voisinage de l'utilisateur, dont les évaluations sont considérées comme similaires aux goûts ou aux besoins de l'utilisateur actuel, par un modèle de contexte social. [6]

	v_1	v_2	v_3	v_4	v_5		u_1	u_2	u_3	u_4	u_5
u_1	5	?	2	?	?	u_1	0	1	0	0	1
u_2	4	4	5	?	?	u_2	0	0	1	1	0
u_3	?	?	4	4	1	u_3	0	0	0	0	0
u_4	?	?	?	?	3	u_4	1	0	1	0	0
u_5	?	?	1	?	?	u_5	0	1	1	1	0

Figure 1.2. Les Relations user-user et user-items[4]

1.2.2. Les Types Des Systèmes De Recommandation Sociaux :

Les systèmes de recommandation sociaux sont appliqués dans plusieurs domaines depuis leurs apparitions, cela a conduit à l'apparition de nombreux types, on site les systèmes les plus courants et les plus utilisés (Contexte, Tag, Trust, Temporal, cross) :

a. SRS Basés Sur Le Contexte (Context-Aware) :

Ces systèmes tenant compte du contexte exploitent les informations contextuelles sur l'utilisateur et l'article recommandé pour générer des recommandations plus personnalisées. Ces

informations contextuelles peuvent inclure le lieu, l'heure, la météo, le dispositif et le contexte social, entre autres. Par exemple, ils peuvent recommander des restaurants à proximité en fonction de l'emplacement d'utilisateur, du jour et de l'heure actuels.[7]

b. SRS Basés Sur Les Mots-Clés (Tag) :

Ces systèmes sont souvent utilisés par les plateformes des médias sociaux, comme Instagram et Pinterest pour recommander du contenu aux utilisateurs en fonction des tags qu'ils ont utilisé ou avec lesquels ils ont interagi. Les tags sont essentiellement des mots-clés qui décrivent le contenu, et les utilisateurs peuvent utiliser les tags pour découvrir du contenu qui correspond à leurs intérêts.[8]

Les sites des réseaux sociaux offrent aux utilisateurs la possibilité de télécharger des informations ou des ressources sous forme d'images, de documents, ou de vidéos. Ces ressources se voient attribuées une étiquette appelée "tag", et la pratique consistant à attribuer des tags est appelée "tagging". Les tags sont des mots personnalisés par les utilisateurs pour exprimer leur opinion, l'emplacement, l'humeur, ou l'heure, etc. Ils servent de lien entre une ressource et un utilisateur, et leur utilisation fréquente démontre l'intérêt d'un utilisateur pour cette ressource. [7]

c. SRS Basés Sur La Confiance (Trust) :

Les SRS basés sur la confiance s'appuient sur le concept de confiance entre les utilisateurs afin de fournir des recommandations. Dans ces systèmes, les utilisateurs ne sont pas seulement caractérisés en fonction de leurs interactions passées avec des articles, mais aussi en fonction de leurs relations avec d'autres utilisateurs.[9]

Le principe est : les utilisateurs sont plus susceptibles de faire confiance aux recommandations d'autres utilisateurs, en qui ils font confiance.

La confiance dans un réseau social dépend des liens sociaux entre les utilisateurs, l'un des principaux avantages de ces systèmes est : ils peuvent aider à maîtriser le problème du démarrage à froid [7], lorsque de nouveaux utilisateurs ou de nouveaux éléments ne disposent que des données d'interaction limitées.

En exploitant les relations de confiance entre les utilisateurs, ces systèmes peuvent fournir des recommandations personnalisées même lorsque les données disponibles sont limitées.

d. SRS Basés Sur Les Données Temporelles (Temporal Data) :

Ces systèmes prennent en compte l'évolution des préférences des utilisateurs au fil du temps. Ces préférences peuvent changer en raison de divers facteurs, tels que des changements dans leurs circonstances de vie, l'exposition à un nouveau contenu, ou l'influence des amis et de

la famille. Le système analyse le comportement et les préférences passées de l'utilisateur pour prédire ses préférences futures. [10]

Les données évoluent dans le temps et les modèles doivent être mis à jour en permanence pour refléter leur nature actuelle. L'analyse de ces données doit trouver un compromis entre l'actualisation des effets temporaires qui ont un impact très faible sur le comportement futur, et la détection des tendances à plus long terme qui reflètent la nature inhérente des données. Ce type de système est couramment utilisé dans les services de streaming musical, où les préférences musicales des utilisateurs peuvent évoluer dans le temps. [10]

e. SRS Basés Sur Les Inter-Domains (Cross Domain) :

Cross-domaine se réfère à l'idée que les intérêts et les préférences des utilisateurs peuvent relever de plusieurs domaines ou catégories, et que des recommandations peuvent être faites sur ces domaines. Les systèmes de recommandation inter-domaines visent à tirer parti du comportement inter domaines des utilisateurs pour fournir des recommandations plus précises et plus variées. [7],[4]

L'un des principaux défis des systèmes de recommandation inter-domaines est la rareté des données "**data sparsity**"¹. Dans de nombreux cas, il peut y avoir un chevauchement limité entre les interactions utilisateur-article dans différents domaines. Par exemple, un utilisateur peut avoir évalué de nombreux livres mais seulement quelques films. Pour relever ce défi, diverses techniques ont été développées, telles que la cartographie des caractéristiques et les approches hybrides. [11]

Ces systèmes visent à résoudre plusieurs problèmes auxquels les systèmes de recommandation traditionnels peuvent être confrontés, notamment la rareté des données, la diversité, le démarrage à froid.

Il existe encore plusieurs types, on a cité les plus utilisés. Les systèmes dont nous avons parlé précédemment exploitent une série de techniques et de sources de données pour fournir des recommandations plus personnalisées et plus pertinentes aux utilisateurs, améliorant ainsi leur engagement et leur satisfaction. Comme le domaine des systèmes de recommandation sociaux continue d'évoluer, de nouvelles approches et techniques sont susceptibles d'émerger, améliorant encore l'efficacité et la rentabilité de ces systèmes.

Data sparsity¹La rareté des données fait référence à une situation où les données disponibles sont incomplètes ou manquent de densité, ce qui peut poser des problèmes pour l'analyse et la modélisation.

1.2.3. Les Méthodes utilisées dans Les SRS :

Un certain nombre de techniques, modèles et approches des SRS a été proposé et mis en œuvre pour effectuer la recommandation des items dans la littérature. Elles sont classées en plusieurs catégories, on cite les suivantes :

a. SRS Basés Sur L'apprentissage Profond (Deep Learning-Based) :

Cette technique utilise des réseaux neuronaux profonds pour apprendre des représentations complexes des utilisateurs et des articles, qui peuvent être utilisées pour générer des recommandations. L'apprentissage profond peut être efficace pour capturer des relations non linéaires et complexes entre les utilisateurs et les articles, ainsi que pour traiter des données hétérogènes et hautement dimensionnelles. Ces systèmes se composent de plusieurs couches de traitement où chaque couche suivante extrait des caractéristiques plus complexes, qui sont agrégées pour être traitées en entrée des couches suivantes.

Les modèles d'apprentissage profond fonctionnent de la même manière que le cerveau humain qui traite les informations et apprend. Ces modèles sont formés à l'aide d'un apprentissage supervisé ou non supervisé. [5] Quelques exemples des modèles utilisés dans l'apprentissage profond (Les réseaux neuronaux profonds, les réseaux neuronaux récurrents, les auto-encodeurs profonds, les réseaux neuronaux convolutionnels ...).

Les modèles d'apprentissage profond peuvent être entraînés sur des ensembles de données à grande échelle, pour apprendre des représentations d'utilisateurs et d'articles qui capturent leurs préférences et leurs caractéristiques. [5]

b. SRS Basés Sur Le Filtrage Collaboratif (Collaborative Filtering-Based) :

Le filtrage collaboratif est une technique courante, utilisée dans les systèmes de recommandation sociale, afin de générer des recommandations personnalisées pour les utilisateurs, en fonction de leur comportement passé et du comportement des autres utilisateurs du système. L'idée derrière le filtrage collaboratif est, que les utilisateurs qui ont des préférences similaires dans le passé sont susceptibles d'avoir des préférences similaires dans le futur. [3]

Le filtrage collaboratif peut être basé sur l'utilisateur ou sur l'article. Dans le cas du filtrage collaboratif basé sur l'utilisateur, la similarité entre les utilisateurs est calculée sur la base de leurs évaluations ou interactions passées avec les articles. les recommandations sont générées sur la base des évaluations des utilisateurs similaires, par contre dans le filtrage collaboratif basé sur les éléments, la similarité entre les éléments est calculée sur la base des évaluations des utilisateurs qui ont interagi avec les deux éléments, et les recommandations sont générées sur la base des évaluations des éléments similaires.[12]

Cette méthode peut améliorer la précision et la diversité des recommandations dans les SRS en tirant parti de la connaissance collective des utilisateurs du système. Cependant, il peut également souffrir du problème du démarrage à froid, lorsque de nouveaux utilisateurs ou des éléments avec peu ou pas d'évaluations n'ont pas d'historique sur lequel baser les recommandations. [12]

c. SRS Hybrides (Hybrid Techniques) :

Les techniques hybrides combinent deux ou plus de deux techniques de façon à ce que les caractéristiques d'une technique puissent corriger les défauts des autres techniques. Généralement, cette technique fusionne les caractéristiques des techniques CBF et CF pour produire de meilleurs résultats. Les recommandations sociales actuelles combinent généralement la technique CF avec d'autres techniques (par exemple, CBF et l'apprentissage profond) afin de donner de meilleurs résultats. [5]

Le choix des techniques de recommandation à combiner dans un SRS hybride dépend du cas d'utilisation spécifique, des données disponibles et des objectifs souhaités pour le système.

Différentes combinaisons de techniques peuvent conduire à des résultats différents, et il est important d'évaluer les performances du système en utilisant des mesures appropriées pour s'assurer que les recommandations sont précises et utiles aux utilisateurs. [13]

d. SRS Basés Sur Les Ensembles Flous (Fuzzy-Based) :

Ces techniques utilisent la logique floue [14] pour modéliser l'incertitude et l'imprécision des préférences de l'utilisateur et générer des recommandations personnalisées, cette théorie des ensembles flous joue un rôle crucial dans la recherche d'information et la prise de décision. Chun Guan et Al ont utilisé le potentiel des ensembles flous intuitionnistes pour représenter des balises incertaines et vagues. DBSCAN (Density-based Spatial Clustering of Applications with Noise) est utilisé pour regrouper les éléments dans lesquels les balises sont caractérisées par des ensembles flous intuitifs. [15]

Les ensembles flous sont connus pour résoudre les problèmes de changements et de comportements dynamiques. Les approches basées sur la logique floue peuvent améliorer la flexibilité et l'adaptabilité du système de recommandation aux préférences variables des utilisateurs. [3]

e. SRS Basés Sur Le Clustering (Clustering-Based):

Les SRS avec clustering sont des techniques qui combinent l'analyse des réseaux sociaux et le clustering pour générer des recommandations personnalisées aux utilisateurs, cette

méthode utilise les approches de similarité pour regrouper les utilisateurs ou les éléments dans différents clusters.

Dans la RS basée sur le clustering, il est facile d'identifier les utilisateurs similaires à l'utilisateur cible, car les goûts et les préférences des utilisateurs appartenant à un cluster sont similaires en fonction du contexte. Par exemple, le regroupement de films basés sur le genre et la recommandation des films qui correspondent au genre spécifié par l'utilisateur. [3]

Les SRS avec mise en cluster peuvent améliorer la précision et la diversité des recommandations. Ils peuvent également atténuer le problème du démarrage à froid en générant des recommandations pour les nouveaux utilisateurs en fonction de leurs attributs et de leurs interactions sociales avec d'autres utilisateurs. Cependant, cette approche nécessite une quantité importante de données et de ressources informatiques pour effectuer le clustering et générer des recommandations. [3]

f. Comparaison des méthodes :

Le Tableau 1.1 présente une comparaison entre les méthodes citées.

METHODES	AVANTAGES	INCONVENIENTS
APPRENTISSAGE EN PROFONDEUR	Traiter de grandes quantités de données complexes. Apprendre automatiquement des représentations de caractéristiques complexes à partir des données.	Nécessite de grandes quantités de données étiquetées pour l'apprentissage. Peut-être coûteux en termes de calcul et nécessite un matériel puissant.
SRS HYBRIDES	Tirer parti des points forts de plusieurs approches de recommandation pour améliorer les performances. Traiter les problèmes de démarrage à froid.	Peut-être plus complexe à mettre en œuvre et nécessite l'intégration de plusieurs systèmes. Nécessite des ressources informatiques supplémentaires
FILTRAGE COLLABORATIF	Fournir des recommandations personnalisées basées sur le comportement et les préférences de l'utilisateur. Peut être adapté à de grands ensembles de données.	Souffrir de problèmes de démarrage à froid. Peut-être biaisé en faveur d'éléments ou d'utilisateurs populaires.
REGROUPEMENT (CLUSTERING)	Regrouper des éléments ou des utilisateurs similaires afin d'améliorer la qualité des recommandations. Traiter des données de retour d'information implicites et explicites.	Souffrir de problèmes de démarrage à froid. Possibilité de ne pas fournir de recommandations personnalisées pour les utilisateurs individuels.
Basé SUR LES ENSEMBLES FLOUS	Gérer l'incertitude et l'imprécision dans les évaluations ou les préférences des utilisateurs/éléments.	Coûteux en termes de calcul pour les grands ensembles de données ou les modèles complexes. Nécessite une expertise dans le domaine

Tableau 1. Les avantages et inconvénients des méthodes citées

1.2.4. Les Défis Et Les Problèmes :

Les systèmes de recommandation sociaux sont confrontés à un certain nombre de défis et de problèmes qui doivent être résolus afin de fournir des recommandations précises et efficaces, surtout avec la taille volumineuse des interactions quotidiennes sur les réseaux sociaux, les SRS sont en train de devenir un cas de recherche sur le Big Data. L'évolution des réseaux est également une orientation prometteuse pour les travaux futurs. Certains des principaux défis et problèmes des SRS sont les suivants :

a. La Diversité :

Le problème de la diversité dans les systèmes de recommandation sociaux concerne le fait que le système fournit des recommandations qui sont trop similaires les unes aux autres, ce qui entraîne un manque de variété dans les choix de l'utilisateur. Cela peut se produire lorsque le système ne recommande que des articles similaires aux préférences passées de l'utilisateur et ignore d'autres articles qui pourraient également l'intéresser.

L'une des raisons de ce problème de diversité est le recours excessif à des algorithmes de filtrage collaboratif qui se concentrent principalement sur les évaluations des utilisateurs et des articles et ne tiennent pas compte d'autres facteurs importants tels que les attributs des articles, les données démographiques des utilisateurs et le contexte. Plusieurs approches ont été proposées pour résoudre le problème de la diversité dans les systèmes de recommandation sociaux, notamment la recommandation basée sur le contenu et la recommandation hybride. [16]

b. Le Problème De Démarrage À Froid (Cold Start) :

Le problème du démarrage à froid dans les SRS fait référence au défi de faire des recommandations précises pour de nouveaux utilisateurs ou de nouveaux articles qui n'ont pas ou ont un très peu de données historiques d'interaction, sans aucune information sur les préférences de l'utilisateur. Donc, il est difficile de fournir des recommandations personnalisées. [17]

Le problème du démarrage à froid peut affecter la performance des SRS, en particulier dans les premières étapes du déploiement ou lors de l'introduction de nouveaux éléments ou utilisateurs. Pour résoudre ce problème, diverses techniques ont été proposées, telles que les recommandations basées sur le contenu, les recommandations basées sur les connaissances et les approches hybrides qui combinent différents types de données et de modèles pour surmonter le manque de données historiques ou même les systèmes de recommandation inter-domaines.[18]

c. La rareté des données (Sparsity) :

Les systèmes de recommandation sociale ont souvent des matrices utilisateur-élément éparses, ce qui signifie que la plupart des utilisateurs n'ont évalué qu'un petit sous-ensemble

d'éléments. Il est donc difficile de modéliser avec précision les préférences des utilisateurs, et cela peut conduire à des recommandations de mauvaise qualité. Pour résoudre le problème de la rareté, certaines méthodes utilisent des techniques de complétion de matrice ou incorporent d'autres sources de données.

Pour résoudre le problème de la rareté, diverses techniques ont été proposées, telles que la factorisation matricielle, les méthodes basées sur le voisinage et le filtrage basé sur le contenu. Ces techniques visent à combler les évaluations manquantes ou à prédire la probabilité qu'un utilisateur interagisse avec un élément sur la base de ses interactions historiques et d'autres informations disponibles. [19]

d. La Confiance :

Les systèmes de recommandation sociale exigent que les utilisateurs fassent confiance au système et aux recommandations fournies. Il peut être difficile d'instaurer la confiance dans le système, surtout lorsque les utilisateurs ont des préférences et des opinions contradictoires

Pour résoudre ce problème, diverses approches ont été proposées[20], comme l'intégration de mesures de confiance dans les algorithmes de recommandation ou la possibilité pour les utilisateurs de mieux contrôler les sources des recommandations qu'ils reçoivent. Cependant, ces solutions nécessitent souvent une conception et une mise en œuvre approfondies afin d'équilibrer les objectifs contradictoires consistant à fournir des recommandations précises tout en respectant la vie privée et l'autonomie des utilisateurs. [21]

e. L'élicitation Des Préférences :

Dans les systèmes de recommandation sociale, le problème d'élicitation des préférences fait référence au défi de capturer et de représenter avec précision les préférences des utilisateurs, en particulier dans les cas où les utilisateurs sont incapables d'articuler clairement leurs préférences ou ont des préférences incomplètes ou incohérentes.

Ce problème découle de plusieurs facteurs : notamment la subjectivité ; la complexité inhérentes aux préférences humaines et le manque de confiance et de transparence dans les réseaux sociaux et la difficulté d'obtenir un retour ou un engagement suffisant de la part d'utilisateur.[22]

1.2.5. Travaux Existants sur Les SRS :

Le domaine des SRS est l'un des domaines de recherche très actifs vu leur émergence et leur utilisation dans la vie quotidienne des personnes. De ce fait, un très grand éventail de travaux de recherche a été proposé ces dernières années sur ces systèmes et leurs méthodes utilisées. Par exemple, dans [23], Katarya et Verma ont suggéré un système de recommandation

de pages web basé sur le clustering fuzzy c-mean. Tandis que, Deng et al. [24] proposent un système de recommandation basé sur la confiance et ils utilisent des encodeurs profonds pour former les caractéristiques initiales cachées des articles et des utilisateurs afin de minimiser la fonction objective. Tandis que, dans [25] les auteurs proposent un système de recommandation sociale hybride qui combine les meilleures caractéristiques de CF et CBF pour recommander des lieux aux utilisateurs, ce système appelé GeoSRS est basé sur des données de géo localisation. Dans [26] un mécanisme adaptatif de sélection des voisins pour la recommandation sociale été introduit. Contrairement à d'autres approches. Shen et al. [27] présentent un système de recommandation sociale basé sur le filtrage collaboratif pour le commerce électronique.

Le tableau 2 résume les travaux présentés dans cette section.

Système	Problème traité	Modèle utilisé
[23]	La classification des utilisateurs, basés sur leur historique de navigation pour créer un système de recommandation basé sur la similarité des utilisateurs.	Un système de recommandation basé sur le clustering flou
DLMF [24]	L'amélioration des recommandations dans les réseaux sociaux en utilisant l'information sur les relations de confiance entre les utilisateurs	Deep autoencoder
GeoSRS [25]	le problème de recommandation de lieux dans un réseau social basé sur la localisation.	Un système de recommandation hybride
SRANS [26]	Sélection des voisins pertinents dans les systèmes de recommandation collaborative. Sélection adaptative des voisins, en utilisant à la fois les informations de notation historique et les informations sociales entre les utilisateurs.	Un système de recommandation adaptatif basé sur la sélection des voisins (Clustering-based)
STR [27]	Le problème traité dans cette étude concerne le développement SR basé sur les données de notation et de relations. L'objectif est de fournir une liste de recommandations personnalisées, en utilisant des modules de calcul de similarité des préférences.	Filtrage Collaboratif avec popularité des items

Tableau 2. Tableau récapitulatif des travaux sur les SRS

1.3. L'Élicitation Des Préférences Dans Les SRS:

1.3.1. Introduction :

Les systèmes de recommandation sociaux sont devenus de plus en plus populaires dans notre vie quotidienne, car ils nous aident à trouver des produits, des services et des contenus qui correspondent à nos intérêts et à nos préférences. Cependant, pour que ces systèmes soient efficaces, ils doivent comprendre les préférences des utilisateurs. L'élicitation des préférences est donc un processus clé pour les systèmes de recommandation sociaux.

L'élicitation des préférences dans les systèmes de recommandation sociaux consiste à comprendre les préférences des utilisateurs en matière de produits, de services et de contenus, afin de leur fournir des recommandations plus pertinentes et personnalisées. Les préférences des utilisateurs peuvent être éliciter à travers différentes méthodes telles que les questionnaires, l'analyse de l'historique de l'utilisateur, l'évaluation de la satisfaction de l'utilisateur et l'analyse de la socialisation.

Cet aspect est important car, avec la croissance exponentielle des données dans les réseaux sociaux, la compréhension des préférences des utilisateurs est plus cruciale que jamais pour offrir des recommandations précises et pertinentes. L'élicitation des préférences des utilisateurs est donc un domaine de recherche actif dans le domaine des systèmes de recommandation sociaux, car elle peut avoir un impact important sur l'expérience utilisateur et sa satisfaction.

Dans cette partie du chapitre, nous présentons les différentes techniques de l'élicitation des références ainsi que les travaux de recherche menés sur les commentaires arabe.

1.3.2. Définition :

L'élicitation des préférences dans les **SRS** est un processus de collecte des préférences ou d'impressions des utilisateurs sur les articles, afin de construire une recommandation personnalisée. Il s'agit d'une étape cruciale du processus de recommandation, car la **précision** et la **pertinence** des recommandations dépendent de la qualité et de précision des données de préférence de l'utilisateur.[28]

Ces informations collecter peuvent être obtenues à partir de sources explicites, telles que les évaluations ou les commentaires fournis par les utilisateurs, ou à partir de sources implicites, telles que le comportement de navigation ou les activités sur les réseaux sociaux. Les techniques d'élicitation de préférences peuvent être basées sur des questionnaires, des algorithmes d'apprentissage automatique ou des modèles de traitement de langage naturel pour extraire des informations des commentaires textuels. Les informations recueillies sont ensuite utilisées pour

fournir des recommandations personnalisées aux utilisateurs, en tenant compte de leurs préférences et de leurs intérêts spécifiques. [29]

Les techniques d'élicitation des préférences peuvent varier en fonction du type de données collectées, du contexte de la recommandation et des préférences de l'utilisateur. Les méthodes courantes comprennent le retour d'information explicite, où les utilisateurs évaluent ou fournissent directement un retour d'information sur les éléments, le retour d'information implicite, qui est déduit du comportement de l'utilisateur et de ses interactions avec le système, et les méthodes hybrides qui combinent le retour d'information explicite et implicite. [28]

Dans l'ensemble, l'élicitation des préférences est un aspect crucial des systèmes de recommandation sociale et joue un rôle important dans l'élaboration de recommandations précises et personnalisées pour les utilisateurs. Dans ce qui suit, nous présentons les méthodes de utilisées pour l'élicitation de préférences.

1.3.3. Les méthodes utilisées dans l'élicitation des préférences :

Dans l'élicitation des préférences, différentes méthodes sont utilisées pour obtenir les préférences des utilisateurs. Voici quelques-unes de ces méthodes :

a. Méthodes Basées Sur Des Questionnaires :

Ces méthodes sont souvent utilisées pour recueillir des informations directes sur les préférences des utilisateurs. Les questionnaires peuvent être conçus de manière à obtenir des informations sur des attributs spécifiques tels que les préférences de prix, les préférences de marques, etc.

Cette méthode est facile à mettre en œuvre, elle peut être utilisée pour obtenir des informations sur un large éventail de critères de préférence, elle permet une personnalisation plus fine des recommandations et elle peut être utilisée pour recueillir des commentaires qualitatifs sur les produits ou services. Cependant, cette méthode peut être coûteuse en termes de temps et d'efforts pour les utilisateurs, et il y a une possibilité de biais de réponse ou de non-réponse si les utilisateurs ne remplissent pas le questionnaire de manière honnête et précise.

De plus, cette méthode a l'avantage de fournir des informations précises sur les préférences des utilisateurs, mais peut être fastidieuse pour l'utilisateur à remplir. [30]

b. Méthodes Bayésiennes :

Les méthodes bayésiennes sont souvent utilisées pour résoudre le problème de l'élaboration de préférences dans les systèmes de recommandation sociaux. Ces méthodes impliquent l'utilisation de modèles probabilistes pour inférer les préférences des utilisateurs en fonction de leurs interactions avec le système de recommandation.

Parmi les méthodes bayésiennes, on peut citer l'approche Bayésienne non paramétrique, qui permet de modéliser les préférences de manière flexible et sans supposition a priori sur la distribution des préférences. L'approche Bayésienne probabiliste est également utilisée couramment, et implique l'utilisation de distributions de probabilité pour modéliser les préférences des utilisateurs.[31]

En général, les méthodes bayésiennes permettent de résoudre efficacement le problème de l'élaboration de préférences dans les systèmes de recommandation sociale, en particulier lorsque les données sont rares ou peu fiables. [31]

c. Méthodes Basées Sur l'analyse des sentiments (Sentiment Analysis) :

Les méthodes d'analyse des sentiments peuvent être utilisées pour obtenir des préférences dans les SRS. Ces méthodes impliquent l'analyse du sentiment ou de l'émotion exprimée dans le contenu généré par les utilisateurs, tel que les critiques de produits, les messages sur les médias sociaux et les commentaires. En analysant le sentiment, il est possible de déterminer la préférence de l'utilisateur pour un produit ou un service particulier.[32]

Il existe plusieurs approches de l'analyse des sentiments, notamment les méthodes statistiques et les méthodes fondées sur l'apprentissage automatique. Les méthodes statistiques impliquent l'utilisation d'algorithmes pour calculer la probabilité qu'un texte exprime un sentiment particulier. Les méthodes basées sur l'apprentissage automatique impliquent l'apprentissage d'un modèle sur un grand ensemble de données d'exemples étiquetés afin de prédire le sentiment de nouveaux morceaux de texte.[33],[32]

Une fois que le sentiment du contenu généré par l'utilisateur a été analysé, il peut être utilisé pour générer des recommandations qui correspondent aux préférences de l'utilisateur. Par exemple, un utilisateur qui exprime un sentiment positif à l'égard d'un type de produit particulier peut se voir recommander des produits similaires à l'avenir. [34]

Il est important de noter que chaque méthode a ses avantages et ses inconvénients, et que la méthode la plus appropriée dépendra du contexte spécifique du système de recommandation.

1.3.4. Les Difficultés et les Problèmes de l'Élicitation des Préférences dans Les SRS :

L'élicitation des préférences dans les Systèmes de Recommandation sociaux (SRS) présente plusieurs difficultés et problèmes, on cite :

a. L'Absence De Feedback Explicite :

Dans les systèmes de recommandation sociale, les utilisateurs peuvent ne pas donner leur avis de manière explicite sur les éléments qu'ils aiment ou n'aiment pas. Il est donc parfois difficile de déduire avec précision leurs préférences, ce qui peut donner lieu à des

recommandations moins personnalisées. Pour relever ce défi, des techniques NLP peuvent être utilisées pour extraire des commentaires implicites de l'activité des utilisateurs sur les médias sociaux. Par exemple, l'analyse des sentiments peut être utilisée pour déterminer si le message ou le commentaire d'un utilisateur reflète un sentiment positif ou négatif à l'égard d'un article ou d'un sujet particulier. [35]

b. Le problème De Démarrage À Froid :

L'un des défis de l'élicitation des préférences est le problème du démarrage à froid, lorsque de nouveaux utilisateurs ou de nouveaux éléments ne disposent d'aucune donnée ou de données très limitées pour la recommandation. Cela peut conduire à des recommandations moins personnalisées et à une mauvaise expérience pour l'utilisateur.

Dans de tels cas, des méthodes alternatives telles que la recommandation basée sur le contenu ou les recommandations sociaux, il est possible aussi d'utiliser des techniques de langage naturel pour exploiter les informations démographiques, telles que l'âge, le sexe ou la localisation, afin de déduire les préférences initiales des nouveaux utilisateurs et de leur fournir des recommandations plus personnalisées. [36]

c. Le Problème De La Confidentialité :

Les utilisateurs peuvent hésiter à partager leurs données personnelles ou leurs préférences sur les réseaux sociaux pour des raisons de confidentialité. Cela peut limiter la quantité de données disponibles pour le système de recommandation et conduire à des recommandations moins précises. Les techniques NLP peuvent être utilisées pour protéger la vie privée des utilisateurs en utilisant des techniques telles que la confidentialité différentielle ou l'apprentissage fédéré, qui permettent d'obtenir des préférences précises tout en préservant la vie privée de l'utilisateur. [37]

d. Le Problème De La Qualité Des Données :

L'un des plus grands défis dans la construction de systèmes de recommandation efficaces est de garantir la qualité des données utilisées pour générer des recommandations. Les problèmes de qualité des données peuvent provenir de divers facteurs, tels que le bruit, le caractère incomplet et l'incohérence des données des utilisateurs. Par exemple, les utilisateurs peuvent fournir un commentaire inexact ou biaisé, ce qui peut conduire à des recommandations de mauvaise qualité.

En outre, certains utilisateurs peuvent être plus actifs que d'autres, ce qui entraîne une distribution inégale des données. Ces problèmes de qualité des données peuvent avoir un impact significatif sur la précision et l'efficacité des algorithmes de recommandation. [38]

e. Le problème De la Diversité :

Dans certains cas, les SRS peuvent recommander des articles trop proches des préférences actuelles de l'utilisateur, ce qui entraîne un manque de diversité dans les recommandations. Cette situation peut être problématique car elle limite l'exposition de l'utilisateur à des éléments nouveaux et potentiellement intéressants.

Par exemple, si un utilisateur écoute fréquemment de la musique classique, un système de recommandation qui ne recommande que d'autres musiques classiques risque de ne pas lui faire découvrir d'autres genres qu'il pourrait apprécier. Ce manque de diversité peut entraîner la lassitude de l'utilisateur et son insatisfaction à l'égard du système de recommandation.[39]

En résumé, l'élicitation des préférences dans les systèmes de recommandation sociale est confrontée à plusieurs défis, pour relever ces défis, il faut combiner des techniques issues de différents domaines, tels que le traitement du **langage naturel**, **l'apprentissage automatique** et **l'analyse des réseaux sociaux**.

1.4. L'élicitation des préférences à partir des commentaires :

L'extraction des préférences à partir des commentaires dans les systèmes de recommandation sociaux est un problème important dans le domaine de la recommandation. Les SRS utilisent les interactions sociales telles que les **commentaires**, les mentions "**j'aime**" et les **partages** pour fournir des recommandations personnalisées aux utilisateurs. Ils fournissent des informations riches sur l'opinion et le sentiment de l'utilisateur à l'égard d'un élément particulier. Ces commentaires peuvent être utilisés pour personnaliser la recommandation pour l'utilisateur en identifiant les aspects de l'article qui sont pertinents pour lui. [40]

Une approche pour obtenir des préférences à partir de commentaires consiste à utiliser des techniques de traitement du langage naturel (NLP) pour extraire des caractéristiques du texte. Les techniques de traitement du langage naturel peuvent être utilisées pour analyser le contenu généré par les utilisateurs, comme les critiques, les commentaires et les messages sur les médias sociaux, afin d'extraire les préférences et les opinions des utilisateurs. Ces techniques permettent au système de comprendre le sens sémantique du texte, ce qui est essentiel pour saisir les nuances des préférences des utilisateurs.

Par exemple, l'analyse des sentiments peut être utilisée pour extraire le sentiment à partir d'un commentaire, qui consiste à analyser le sentiment ou l'émotion exprimée dans un texte. Cette analyse peut être utilisée pour déterminer si un utilisateur aime ou n'aime pas un élément particulier, ce qui peut servir d'entrée au système de recommandation. [41]

Enfin, certains systèmes de recommandation sociaux utilisent des techniques de filtrage collaboratif pour recommander des articles sur la base des préférences d'autres utilisateurs qui ont des modèles de commentaires ou des préférences similaires.

Dans l'ensemble, l'obtention de préférences à partir de commentaires est un problème important pour les systèmes de recommandation sociaux et nécessite une combinaison des techniques de traitement du langage naturel, d'apprentissage automatique et de filtrage collaboratif...etc.

1.5. Travaux existants sur l'élicitation des préférences à partir des commentaires arabes :

Dans le domaine de l'élicitation des préférences à partir des commentaires en arabe, plusieurs travaux récents ont été réalisés pour comprendre les choix et les préférences des utilisateurs arabophones.

Voici quelques exemples de travaux récents sur l'élicitation des préférences à partir des commentaires en arabe dans le tableau 3 :

TITRE DU TRAVAIL	TACHE	APPROCHE
arHateDetector : Détection des discours haineux à partir de tweets arabes[42]	Détection des discours haineux	Techniques d'apprentissage profond pour la classification
Classification des émotions dans la poésie arabe à l'aide de l'apprentissage profond[43]	Classification des émotions dans la poésie arabe	Algorithmes d'apprentissage profond pour la classification des émotions
Approche basée sur AraBERT et la segmentation Farasa[44]	Détection du sarcasme et des sentiments	Utilisation du modèle AraBERT et la segmentation par Farasa pour la détection du sarcasme et des sentiments
Classification des secteurs de télécommunications en Arabie saoudite.[45]	Classification des secteurs	Les techniques de classification pour les secteurs de télécommunications en Arabie saoudite
Analyse des sentiments des commentaires YouTube en arabe[46]	Analyse des sentiments	Analyse des sentiments sur les commentaires YouTube en arabe
Analyse des sentiments des commentaires argotiques arabes sur Facebook[47]	Analyse des sentiments	Analyse Des sentiments dans les commentaires argotiques arabes sur Facebook

Tableau 3. Tableau récapitulatif des travaux sur l'élicitation des préférences à partir des commentaires arabes

D'après les travaux présentés, on a constaté que modèle AraBert a prouvé une efficacité immense lors de son utilisation pour le traitement des commentaires en Arabe, ce qui nous a motivé à l'utiliser pour faire l'élicitation des préférences dans notre système de recommandation.

Avant de passer à la partie conception et afin de donner au lecteur une idée sur cet outil car il est nouveaux, nous présentons dans la section suivante un aperçu sur le modèle AraBert.

1.6. Le modèle AraBERT :

Arabert[W4][44] est un modèle de langage pré-entraîné développé par le Centre d'excellence en IA pour l'Arabie saoudite de l'Université des sciences et technologies du roi Abdallah (KAUST), spécifiquement pour le texte arabe. Le modèle est fondé sur l'architecture BERT (**Bidirectional Encoder Representation from Transformer**), c'est un réseau de neurones basé sur un transformateur, et conçu pour le pré-entraînement de grandes quantités de données textuelles. Arabert diffère des autres modèles de langage en cela qu'il est spécifiquement optimisé pour traiter la morphologie complexe et les signes diacritiques du texte arabe. [34]

Arabert peut être utilisé pour diverses tâches de NLP, telles que la classification de textes, la reconnaissance d'entités nommées, l'analyse de sentiments (SA), la traduction automatique et la réponse à des questions. L'un des principaux avantages d'Arabert est sa capacité à traiter avec précision la morphologie complexe et les signes diacritiques du texte arabe, ce qui peut constituer un défi pour d'autres modèles de langage.

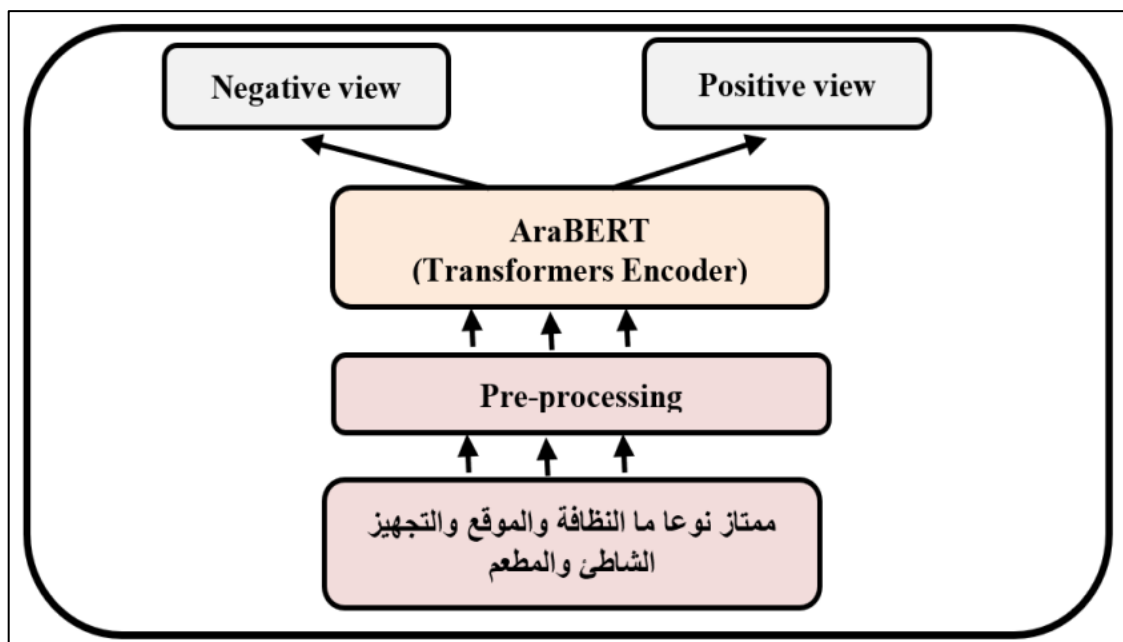


Figure 1.3. l'architecture de AraBERT.[48]

1.6.1. Les entrées du modèle :

Arabert est un modèle basé sur un transformateur, il n'utilise que la partie encodeur, il a été pré-entraîné sur un grand corpus de texte arabe en utilisant l'objectif de modélisation du langage masqué. L'architecture du modèle peut être divisée en deux composants principaux : la couche d'entrée et les couches de transformation. [2]

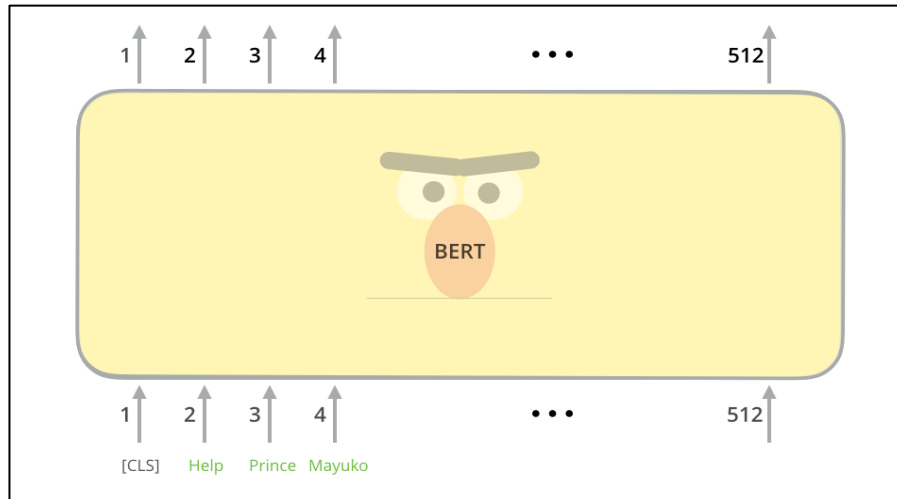


Figure 1.4. L'entrée du modèle arabert. [W2]

La couche d'entrée d'Arabert se compose d'une couche d'intégration suivie d'une couche d'encodage positionnel. La couche d'intégration associe chaque jeton d'entrée à une représentation vectorielle dense. Ces encastresments sont appris au cours de la formation et capturent le sens et les informations contextuelles des mots. La couche d'encodage positionnel ajoute des informations sur les positions relatives des mots dans la séquence d'entrée.

```
Texte d'origine: بسم الله الرحمن الرحيم
Jetons: ['[CLS]', 'رحيم##', 'ال', 'ن##', 'رحم##', 'ال', 'الله', 'بسم', '[SEP]']
IDs d'entrée: [29756, 3138, 12695, 445, 28157, 1003, 445, 44540, 29758]
Masque d'attention: [1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Figure 1.5. Résultat de pré-traitement de texte

1.6.2. Tokenisations :

Contrairement à BERT qui utilise la méthode de tokenisation WordPiece, Arabert utilise la bibliothèque Farasa pour la segmentation des mots.

Farasa est une bibliothèque de traitement de la langue arabe open-source développée par l'Université King Saud en Arabie Saoudite. Cette bibliothèque est conçue pour résoudre les défis spécifiques associés à la langue arabe, tels que la segmentation des mots, la suppression des préfixes et des suffixes, la normalisation des caractères et la résolution des ambiguïtés morphologiques. [49]

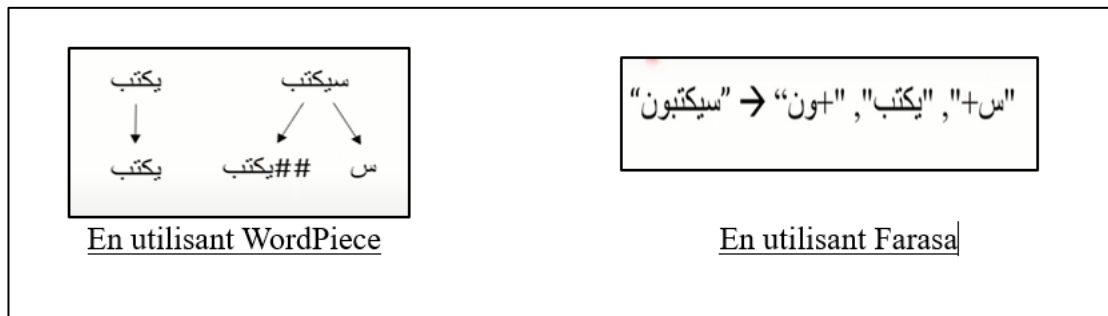


Figure 1.6. La différence entre la segmentation de texte avec (Wordpiece et Farasa)

Les couches de transformation constituent le cœur du modèle, elles sont responsables de l'apprentissage des représentations contextuelles des mots d'entrée. Elles se composent de plusieurs sous-couches qui effectuent des opérations spécifiques. Chaque couche de transformation se compose de deux sous-couches : une couche d'auto-attention à plusieurs têtes et une couche d'anticipation. [50]

- **La couche d'auto-attention :**

La couche d'auto-attention à plusieurs têtes calcule les poids d'attention entre tous les jetons d'entrée, puis regrouper les jetons sur la base de ces poids. Le mécanisme d'auto-attention permet au modèle de s'intéresser aux différentes parties de la séquence d'entrée, et d'apprendre des informations contextuelles sur chaque jeton. L'aspect multi têtes de cette couche permet au modèle de s'intéresser à la séquence d'entrée de différentes manières, améliorant ainsi sa capacité à capturer les dépendances complexes entre les mots-clés. [50]

- **La couche d'anticipation : (feed-forward layer)**

Applique une couche entièrement connectée par points avec une fonction d'activation ReLU à la sortie de la couche d'auto-attention. Cette couche est chargée de capturer les relations non linéaires entre les mots et d'extraire des caractéristiques plus complexes de la séquence d'entrée. [38]

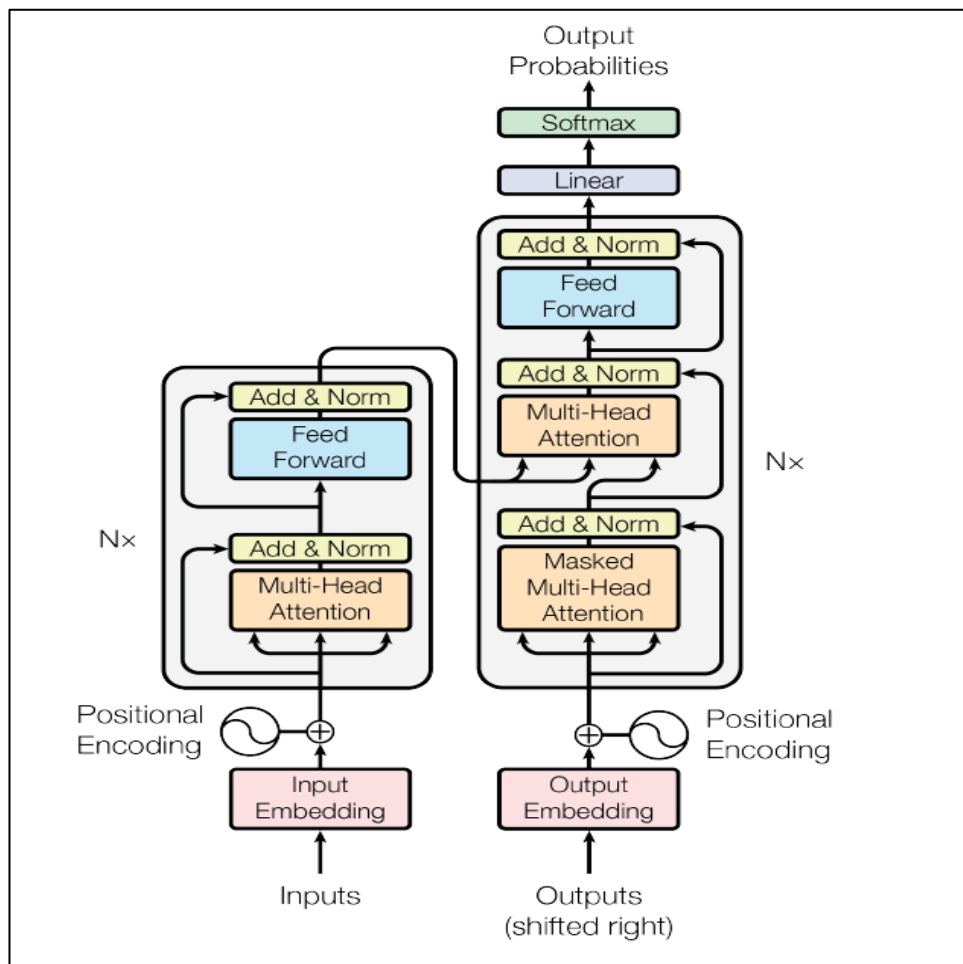


Figure 1.7. L'Architecture du model Transformer. [W2]

La sortie des couches de transformation passe ensuite par une couche de mise en commun qui regroupe les représentations de tous les **tokens** d'entrée en un seul vecteur. La couche de mise en commun² peut être une couche de mise en commun moyenne ou maximale. Ce vecteur est ensuite introduit dans une couche de classification, qui prédit l'étiquette ou la sortie de la séquence d'entrée. [50].

- **Arabert est disponible en deux versions** : version de base (Arabert-base) et version large (Arabert-large).

La version de base utilise 12 couches de transformation et a un total de 110 millions de paramètres entraînaables, tandis que la version large utilise 24 couches de transformation et a un total de 320 millions de paramètres entraînaables.

²La couche de mise en commun, également appelée couche de pooling, est une étape dans les modèles de traitement du langage naturel où les représentations des tokens d'entrée sont agrégées en un seul vecteur.

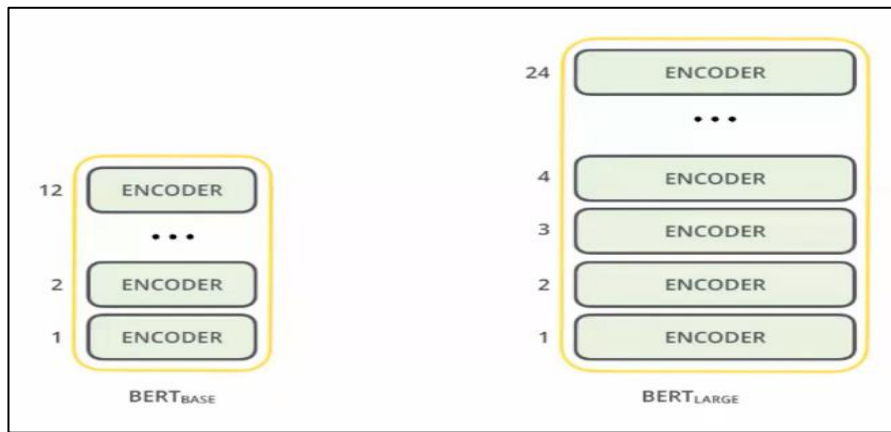


Figure 1.8. Versions de model Arabert (BASE et LARGE).[W2]

En raison de sa taille grande, la version large est capable d'apprendre des représentations plus riches et complexes, ce qui peut conduire à de meilleures performances dans certaines tâches de traitement du langage naturel.[2]

1.6.3. Les procédures de pré-entraînement et le Réglage fin de BERT :

a. Pré-entraînement (Pre-training) :

Conformément à l'objectif initial de pré-entraînement Arabert, une tâche de modélisation du langage masqué (MLM) a été utilisée en masquant 15% des **tokens** d'entrée, où les mots entiers sont remplacés 80% du temps par le jeton [MASK], 10% par un jeton aléatoire, et 10% par le jeton original. En utilisant le masquage des mots entiers, le modèle est forcé de prédire le mot entier plutôt que seulement les parties du mot. En plus de cela, la tâche de prédiction de la phrase suivante (NSP³) a également été utilisée, ce qui aide le modèle à comprendre la relation entre deux phrases, et qui peut être utile pour de nombreuses tâches de compréhension du langage telles que la réponse aux questions.[1]

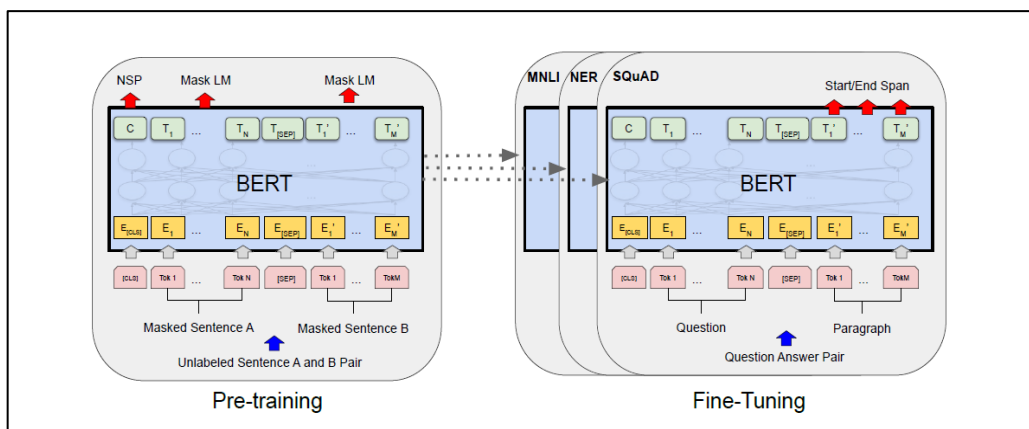


Figure 1.9. Les procédures de pré-entraînement et de Réglage fin de Arabert. [W2]

³NSP signifie Next Sentence Prediction (prédiction de la phrase suivante). Il s'agit d'une tâche utilisée pendant la phase de pré-entraînement des modèles de langage, y compris Arabert. L'objectif de la tâche NSP est d'entraîner le modèle à prédire si deux phrases d'entrée apparaissent consécutivement ou non.

b. Réglage fin (Fine-tuning) :

Pour la classification de séquences avec AraBERT, la dernière couche cachée de la première unité est utilisée comme représentation de l'ensemble de la phrase. Cette couche cachée correspond à la représentation du jeton spécial "[CLS]" ajouté au début de chaque phrase.

Ensuite, une simple couche d'anticipation est ajoutée avec une fonction Softmax pour obtenir une distribution de probabilité sur les classes de sortie. Lors du réglage fin, le classificateur et les poids du modèle pré-entraîné sont entraînés simultanément pour maximiser la probabilité logarithmique de la classe correcte. [50] , [1]

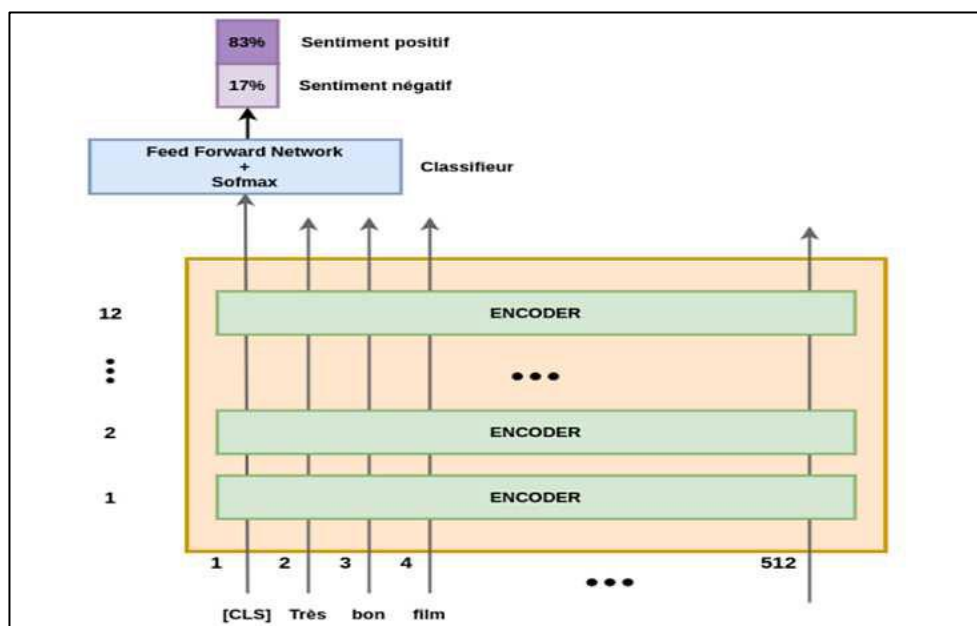


Figure 1.10. Architecture du réglage fin (Fine-tuning). [W2]

En plus de la classification des séquences, AraBERT peut également être affiné pour d'autres tâches de traitement de langage naturel telles que la reconnaissance d'entités nommées (Named Entity Recognition, NER) et la réponse aux questions (Question Answering, QA).

1.7. Conclusion :

Ce premier chapitre a permis de réaliser une évaluation de l'état de l'art des systèmes de recommandation sociaux (SRS). Nous avons commencé par présenter le sujet et les nombreuses définitions liées aux SRS. Ensuite, nous avons étudié les différents modèles des SRS.

En ce qui concerne les méthodologies utilisées dans le SRS, nous avons examiné plusieurs approches, afin de mettre en évidence les caractéristiques uniques de chaque méthode, poursuit par une comparaison entre elles.

Nous avons également abordé les défis et les problèmes rencontrés dans le cadre du SRS, ainsi que les nombreuses techniques utilisées pour obtenir les préférences. Ensuite, nous avons concentré sur l'obtention de préférences basées sur les commentaires des utilisateurs et nous avons souligné l'importance de cette stratégie en donnant un aperçu des travaux connexes déjà réalisés. Enfin, nous avons présenté le modèle Arabert, qui est un modèle de traitement du langage naturel basé sur BERT.

Dons, ce premier chapitre a fourni un aperçu approfondi des systèmes de recommandation sociaux et de l'élicitation des préférences. Il a jeté les bases nécessaires pour la conception de notre système en utilisant le modèle Arabert. Dans le prochain chapitre, nous détaillerons notre système et sa conception en exploitant les capacités du modèle Arabert.

Pour plus d'information consulter [l'annexe A](#).

Chapitre 2 : Conception et implémentation

2.1. Introduction

Dans ce chapitre, nous proposons une solution pour l'analyse des sentiments dans le cadre des systèmes de recommandation sociaux. Notre objectif est d'utiliser le modèle Arabert, un modèle de deep learning pour le traitement automatique du langage naturel en langue arabe, afin de réaliser une analyse précise des sentiments exprimés par les utilisateurs dans leurs commentaires.

2.2. Contexte

Les réseaux sociaux sont devenus des plateformes de choix pour la communication, la socialisation et le partage de contenu dans notre vie quotidienne. Récemment, un grand nombre d'utilisateurs de ces réseaux favorisent l'utilisation de la langue arabe pour donner leurs avis sur les produits, commenter des publications...etc. Cependant, les utilisateurs arabophones sont souvent confrontés à des défis uniques lorsqu'ils interagissent avec les réseaux sociaux pour en trouver le contenu adéquat à leurs préférences et intérêts, notamment en ce qui concerne la langue arabe complexe et les différences culturelles.

Donc, la mise en place d'un système de recommandations social efficace est devenue une priorité pour améliorer l'expérience de l'utilisateur et augmenter la fidélité des utilisateurs arabophones, vu l'importance croissante des réseaux sociaux dans la vie quotidienne des utilisateurs.

Par conséquent, l'utilisation des modèles de l'élicitation des préférences depuis les commentaires en arabe dans un SRS est donc de plus en plus nécessaire afin de comprendre les préférences des utilisateurs arabophones et fournir des recommandations personnalisées pertinentes.

2.3. Problématique

Le traitement des commentaires en arabe présente des défis majeurs en termes d'obtention des préférences précises des utilisateurs. La langue arabe possède un vocabulaire riche et une grammaire complexe, avec divers dialectes et variations régionales. Il est donc difficile d'interpréter avec précision les commentaires des utilisateurs. En outre, les différences culturelles et les nuances dans l'utilisation de la langue peuvent compliquer d'avantage le processus de compréhension des préférences des utilisateurs.

Donc, la problématique de ce travail de recherche s'articule autour de deux axes principaux :

- **Comment extraire efficacement les préférences des utilisateurs à partir des commentaires arabes dans les réseaux sociaux ?**
- **Comment faire une recommandation personnalisée aux utilisateurs sur la base de ces préférences extraites ?**

2.4. Solution Proposée :

Pour répondre à ces questions, l'étude présentée dans ce manuscrit vise à explorer différentes méthodes d'élicitation des préférences qui peuvent être utilisées pour comprendre les commentaires arabes sur les réseaux sociaux.

Ensuite, ces préférences seront employées dans le processus de la recommandation afin d'utiliser les méthodes de filtrage collaboratif et de filtrage basé sur le contenu, traditionnelles afin d'obtenir des recommandations pertinentes et personnalisées.

Pour résoudre les défis de traitement des commentaires arabes et extraire les préférences des utilisateurs, on propose d'utiliser le modèle **Arabert** qui est une variante du modèle populaire de traitement du langage naturel **BERT** (Bidirectional Encoder Representations from Transformers) qui a été spécifiquement entraîné sur des données en langue arabe.[2]

Arabert utilise des techniques avancées de modélisation de langue naturelle, telles que le traitement de séquence à séquence (seq2seq), l'auto-encodage et la prédiction de masquage de token (tokenmasking) pour apprendre des représentations riches de la langue arabe. Comme BERT, Arabert est un modèle basé sur des transformateurs qui peuvent être affinés pour diverses tâches de traitement du langage naturel, telles que la classification linguistique, la réponse aux questions. Il s'est avéré qu'il obtenait des résultats de pointe sur plusieurs critères d'évaluation de la langue arabe, démontrant ainsi son efficacité à capturer les caractéristiques linguistiques complexes de la langue.[2] Cet outil a été mis à la disposition de la communauté des chercheurs sous une licence open-source⁴, ce qui permet aux chercheurs et aux développeurs d'utiliser le modèle et de s'en inspirer pour créer des applications innovantes de traitement de la langue arabe.

Dans les sections qui suivent, nous présentons en détails les différentes étapes de la conception et de la réalisation de notre système de recommandation basé sur l'élicitation des préférences à partir des commentaires arabes (*Arabic Preference elicitation based Recommender ArbPrefRec*), y compris les méthodologies et techniques utilisées.

⁴<https://github.com/aub-mind/arabert>

2.5. Motivation et objectifs

Le Traitement Automatique du Langage Naturel (NLP) est un domaine clé de l'apprentissage automatique (Machine learning) qui vise à donner aux ordinateurs la capacité de comprendre et de traiter le langage humain en utilisant des algorithmes de traitement du langage naturel.

En octobre 2018, la filière d'intelligence artificielle de Google (Google AI) a publié un modèle de deep learning pré-entraîné, appelé BERT, capable de résoudre plusieurs problématiques de NLP. En 2019, la communauté de recherche en NLP a vu l'émergence d'un modèle de deep learning de pointe pré-entraîné appelé Arabert, conçu spécifiquement pour le traitement du langage naturel en langue arabe. Arabert est considéré comme le modèle le plus performant pour le traitement du langage naturel en arabe. Il est important de noter que Arabert est une extension de BERT, utilisant la même architecture de base, mais entraîné sur des données arabes spécifiques, ce qui lui confère une meilleure performance dans le contexte de la langue arabe. Ainsi, Arabert peut être considéré comme une adaptation de BERT pour le NLP en langue arabe.[48]

Dans ce travail, nous visons à utiliser le modèle de langage Arabert pour effectuer un traitement efficace des commentaires des utilisateurs arabophones dans le contexte des systèmes de recommandation sociaux. L'objectif principal est de capturer les sentiments des utilisateurs et de déduire leurs préférences à partir de leurs commentaires. De plus, nous espérons obtenir des résultats précis et fiables, ce qui améliorera considérablement la qualité des recommandations fournies par les systèmes de recommandation dans les réseaux sociaux.

En utilisant le modèle Arabert, on vise à développer des méthodes efficaces pour le traitement de texte arabe, l'extraction de sentiments et d'émotions des commentaires et l'identification de préférences des utilisateurs. Ces méthodes peuvent, ensuite, être utilisées pour développer un système de recommandation social qui personnalise les recommandations en fonction des préférences de chaque utilisateur individuel.

2.6. Conception du système « ArbPrefRec » :

Dans ce qui suit, nous décrivons les détails de la conception et l'implémentation du système "ArbPrefRec" pour l'élicitation des préférences et la génération de recommandations personnalisées.

Nous avons suivi plusieurs étapes, dont l'étape 1 comprend la préparation des données, la configuration du modèle Arabert et l'apprentissage/évaluation du modèle et l'étape 2 explore la

recommandation hybride, collaborative et basée sur le contenu, renforcée par les préférences élicitées afin de générer des recommandations personnalisées. Les différentes étapes sont mentionnées dans la Figure 2.1.

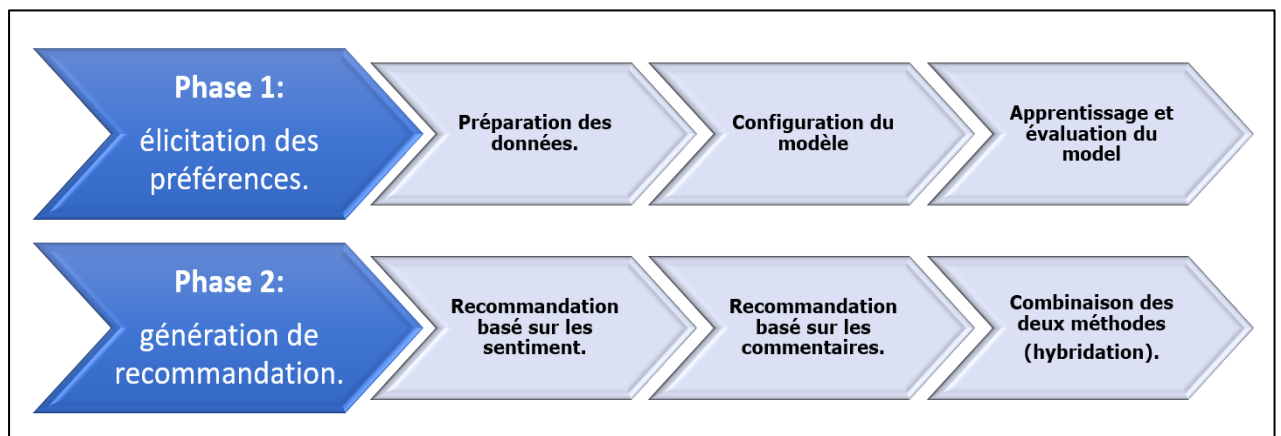


Figure 11. Organigramme représentatif des processus de réalisation de système ArbPrefRec.

Dans ce qui suit, chaque étape du processus sera bien expliquée et détaillée.

2.6.1. ETAPE 1 : L'Élicitation Des Préférences :

Dans cette étape, on vise l'extraction des sentiments des utilisateurs soit (positifs ou négatifs) à partir de chaque commentaire généré par l'utilisateur. Il est important que notre système soit performant et précis parce que ces informations vont jouer un rôle important dans la génération des recommandations dans l'étape suivante. Les différentes étapes de la phase d'élicitation des préférences sont mentionnées dans la Figure 2.2.

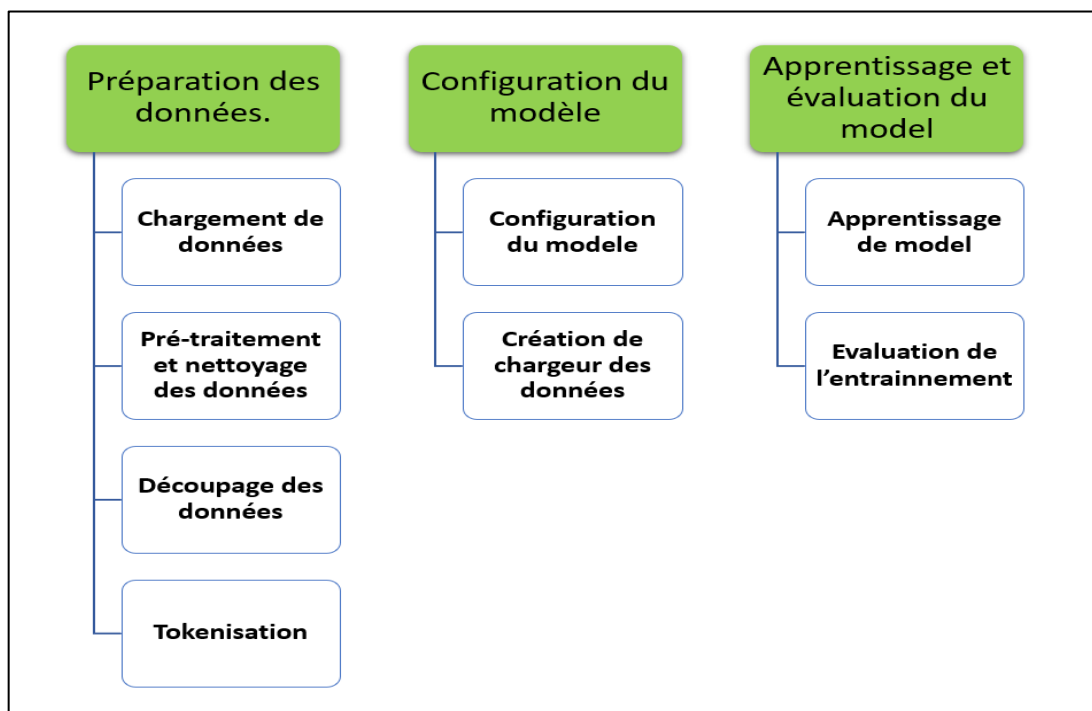


Figure 12. Les étapes de l'élicitation des préférences dans le système ArbPrefRec.

A noter que ces étapes seront exécutées successivement.

Afin de comprendre leur fonctionnement, une explication détaillée est présentée dans ce qui suit :

a. Préparation Des Données :

Tout d'abord, on commence par tout ce qui concerne le nettoyage des données et leur préparation pour qu'elles soient prêtes à être utilisées par le système afin d'appliquer l'apprentissage du modèle.

➤ Chargement de Dataset :

Dans cette étape, on commence par l'importation de la bibliothèque Pandas et le chargement du fichier CSV contenant notre Dataset. Ensuite, on utilise la méthode "head()" pour afficher les premières lignes du Dataset et s'assurer que le chargement s'est bien déroulé.

• Exemple de code source :

```
1 import pandas as pd
2 df = pd.read_csv('/content/drive/MyDrive/ar_res_csv_reviews.csv')
3 df.head()
```

	polarity	text	restaurant_id	user_id
0	0	...اولا: المنيو تغير الشورما اصبحت اعتياديه بأختف	296	423
1	0	...من محلات الشاورما ذات الشعبية لتمييز الصلصات ال	296	423
2	1	...دجاج طازج يحضر امامك على الطلب لا يقوم باعادة ا	5027	39580
3	1	...فكما تعرف استراليايا بالكنغر والكوالا. فإنها تعر	642	444
4	0	...إسمحو لي أن أقيم مطعم هاشم بصفتي فلسطيني عشت ف	434	2191

Figure 13. Chargement et affichage des premières lignes du Dataset.

Une fois que cette étape soit validée, on peut passer aux étapes de prétraitement, de nettoyage et de codage afin de préparer l'ensemble de données.

➤ Pré-traitement et nettoyage des données :

Dans l'analyse de texte, l'une des étapes clés est le prétraitement des données textuelles. Cela implique une série d'étapes qui doivent être appliquées pour nettoyer et préparer les données pour le modèle. Pour cela, les données doivent être soumises à un certain nombre de procédures.

L'utilisation de la bibliothèque "re" avec des expressions régulières pour éliminer les émojis inutiles du texte. Les émojis peuvent également contenir des informations non pertinentes qui peuvent rendre l'analyse des données plus complexe, quelle que soit la langue utilisée.

De plus, en utilisant des expressions régulières, nous pouvons également détecter et supprimer les hyperliens et les mots anglais qui peuvent apparaître dans les commentaires en arabe. Cela nous permettra de simplifier et nettoyer les données textuelles pour l'analyse.

Ensuite on a utilisé une nouvelle bibliothèque dédiée au traitement du langage Arabe appelée "Farasa⁵". Avec cette bibliothèque, les mots et les phrases sont segmentées afin d'aider à structurer les données. De cette manière, le texte est divisé en unités plus petites pour une meilleure analyse et organisation. En fin de compte, le processus de segmentation du texte est un outil utile pour la gestion des données.

- Exemple illustratif :

```
[ ] 1 df.head()
```

	polarity	text	restaurant_id	user_id
0	0	أول+ال+منيو تغير ال+شورما أصبح+ت اعتيادي+ه ب	296	423
1	0	من محل+ات ال+شاورما ذات ال+شعبيه ل+تميز ال+صلص	296	423
2	1	+دجاج طازج يحضر امام+ك على ال+طلب لا يقوم باعد	5027	39580
3	1	...ف+ك+ما تعرف أسترااليا ب+ال+كتغر و+ال+كوالا ف+إن	642	444
4	0	...إسحو لي أن أقيم مطعم هاشم ب+صف+ت+ي فلسطيني عش	434	2191

Figure 14. Le jeu de données après le nettoyage et le prétraitement.

➤ Découpage des données :

Dans cette étape, on a divisé l'ensemble de données initial (`df`) en un ensemble d'entraînement, ensemble de test, avec une distribution de 80:20 respectivement. Ensuite, on a divisé l'ensemble de test, précédemment créé, en un ensemble de validation et un ensemble de test, avec une distribution de 50:50. Cette technique est essentielle pour évaluer les performances d'un modèle d'apprentissage automatique et éviter le sur-apprentissage (**overfitting**).

Le code ci-dessus utilise la fonction `train_test_split` de la bibliothèque 'scikit-learn' en Python pour diviser un ensemble de données en deux parties :

```
train_df, test_df = train_test_split(df, test_size=0.2)
val_df, test_df = train_test_split(test_df, test_size=0.5)
```

Figure 15. Division des données en ensembles d'entraînement et de test.

⁵Farasa est une boîte à outils de traitement du langage naturel (NLP) arabe développée par le groupe de recherche ArabicLanguage Technologies (ALT) du Qatar Computing Research Institute (QCRI). Il fournit une gamme d'outils et de ressources NLP spécifiquement conçus pour le traitement de textes arabes.

On a utilisé la bibliothèque *datasets* de *HuggingFace* pour convertir nos *DataFrames Pandas* en objets *Dataset*. Ensuite on a fusionné les objets *Dataset* en un seul dictionnaire *DatasetDict* contenant les ensembles des données d'apprentissage, de test et de validation.

En utilisant le dictionnaire *DatasetDict*, on obtient une structure des données organisée et pratique pour les gérer, ce qui facilite les étapes suivantes de tokenisation, d'entraînement et d'évaluation du modèle. La Figure 2.6 présente le code source de la création du dictionnaire *DatasetDict*.

```
1 dataset_dict
DatasetDict({
  train: Dataset({
    features: ['polarity', 'text', 'restaurant_id', 'user_id'],
    num_rows: 6691
  })
  valid: Dataset({
    features: ['polarity', 'text', 'restaurant_id', 'user_id'],
    num_rows: 836
  })
  test: Dataset({
    features: ['polarity', 'text', 'restaurant_id', 'user_id'],
    num_rows: 837
  })
})
```

Figure 16. Affectation des données au dictionnaire *DatasetDict*.

➤ **Tokenisation :**

Dans cette étape, le texte sera converti en une séquence de jetons à l'aide du **tokenizer** du modèle "**aubmindlab/bert-base-arabertv01**" entraîné sur des données en arabe. Le tokenizer divise chaque phrase en jetons, en utilisant des techniques de sous-échantillonnage pour gérer la variabilité de la longueur des phrases.

Il est important de définir correctement les noms des colonnes pour les entrées et les étiquettes de notre jeu de données. En effet, les modèles de **HuggingFace's**⁶ de **transformers** s'attendent généralement à des entrées appelées "**input_ids**", et "**attention_mask**", ainsi qu'une colonne d'étiquettes appelée "**labels**".

La figure 2.7 présente le contenu de dictionnaire après la tokenisation.

⁶HuggingFace's est une bibliothèque et une plateforme open-source pour le traitement du langage naturel (NLP). Il offre une collection complète de modèles pré-formés, y compris BERT, GPT, et RoBERTa, qui sont largement utilisés pour diverses tâches NLP.

```

1  tokenized_datasets

DatasetDict({
  train: Dataset({
    features: ['polarity', 'text', 'restaurant_id', 'user_id', 'input_ids', 'token_type_ids', 'attention_mask'],
    num_rows: 6691
  })
  valid: Dataset({
    features: ['polarity', 'text', 'restaurant_id', 'user_id', 'input_ids', 'token_type_ids', 'attention_mask'],
    num_rows: 836
  })
  test: Dataset({
    features: ['polarity', 'text', 'restaurant_id', 'user_id', 'input_ids', 'token_type_ids', 'attention_mask'],
    num_rows: 837
  })
})

```

Figure 17. le contenu de dictionnaire après la tokenisation.

Exemple illustratif :

- Original Text: هذا المطعم حقا رائع انصح الجميع بتجربته
- Segmented Text: هذا ال+مطعم حق+ا رائع انصح ال+جميع ب+تجرب+ت+ه
- Tokenized Text: ["[CLS]", ' ', '+', 'ب', 'جميع', '+', 'ال', '+', 'رائع', 'انصح', '+', 'ال', '+', 'مطعم', '+', 'هذا', '+', 'تجرب', '+', 'ت', '+', '[SEP]']
- Token IDs : [33, 387, 288, 45, 6453, 584, 45, 139, 3428, 26577, 288, 45, 630, 140, 45, 1944, 45, 142, 45, 165, 34]

b. Configuration Du Modèle Arabert :

Dans cette section, nous présentons le modèle Arabert, qui est un modèle pré-entraîné pour le traitement du langage naturel en arabe.

Tout d'abord, on a chargé la configuration du modèle **Arabert** à l'aide de la méthode **from_pretrained** de la bibliothèque **AutoModelForSequenceClassification**.

Ensuite, on a modifié certains paramètres du modèle pour répondre à nos besoins. Au début, on a entraîné le modèle avec sa configuration par défauts, puis, on a ajusté le nombre de couches cachées pour le rendre égale à 12, le nombre de têtes d'attention (attention head) pour le mettre à 12.

Le Figure 2.8, présente le code source qui permet la configuration de modèle.

```

Model config BertConfig {
  "_name_or_path": "aubmindlab/bert-base-arabertV01",
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "transformers_version": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 64000
}

```

Figure 18. La configuration du modèle utilisé dans le système ArbPrefRec.

Après cette étape, on passe à la création du chargeur de données qui est responsable de la mise en lots des exemples de données pour l'entraînement du modèle

➤ **Création de chargeur :**

Dans cette étape, nous avons utilisé le chargeur de données **DataCollatorWithPadding** de la bibliothèque **Transformers** pour l'entraînement de notre modèle. Ce chargeur assure que toutes les séquences d'entrée ont la même longueur en ajoutant des **tokens de remplissage (padding)** au besoin, ce qui permet de maximiser l'efficacité du traitement parallèle et d'accélérer l'entraînement.

Le **DataCollatorWithPadding** prend également en charge le **masquage de tokens (masking)** pour l'entraînement avec la méthode *MaskedLanguageModeling (MLM)* de BERT. En utilisant le **DataCollatorWithPadding**, on a pu entraîner notre modèle sur des données en entrée de longueurs variables avec une efficacité maximale.

La Figure 2.9, présente le code source permettant la création du chargeur de données.

```

from transformers import DataCollatorWithPadding
data_collator = DataCollatorWithPadding(tokenizer)

```

Figure 19. Chargeur de données utilisé.

c. Apprentissage et Evaluation Du Modèle :

On a utilisé l'api de **pytorch** de la bibliothèque **transformer** pour tout ce qui concerne l'entraînement, et l'évaluation durant cette étape.

➤ Apprentissage de modèle :

Nous avons entraîné notre modèle en utilisant le module **Trainer** de **PyTorch**. Nous avons configuré notre entraînement en spécifiant les hyper-paramètres tels que le nombre d'époques, la taille du batch, le taux d'entraînement (**learning rate**).

Nous avons également utilisé le chargeur de données **datacollator** que nous avons créé pour grouper les données en **batches** et les préparer pour l'entraînement. Enfin, nous avons lancé l'entraînement en appelant la méthode **train ()** du **Trainer**.

La Figure 2.10 présente le code source qui permet le réglage des paramètres du modèle dans la phase d'entraînement.

```

# Définir vos arguments d'entraînement
training_args = TrainingArguments(
    output_dir="./output",
    num_train_epochs=10,
    per_device_train_batch_size=40,
    per_device_eval_batch_size=40,
    learning_rate=3e-5,
    weight_decay=0.01,
    evaluation_strategy="epoch",
    logging_steps=100,
)
1 # Créer une instance du Trainer
2 trainer = Trainer(
3     model=model,
4     args=training_args,
5     train_dataset=train_dataset,
6     eval_dataset=eval_dataset,
7     data_collator=data_collator,
8     tokenizer=tokenizer,
9     compute_metrics=compute_metrics,
10 )
11 # Lancer l'entraînement
12 trainer.train()

```

Figure 20. Réglage des paramètres d'entraînement.

Pendant l'entraînement, le modèle est mis à jour en fonction des prédictions et l'exactitude calculées par la fonction de perte. L'entraînement est effectué sur différentes époques jusqu'à ce que la valeur de perte diminue.

La Figure 2.11 présente les différentes valeurs d'évolution de la performance du modèle.

Epoch	Training Loss	Validation Loss	Accuracy
1	0.425400	0.288607	0.889952
2	0.272600	0.273221	0.903110
3	0.192800	0.282970	0.892344
4	0.130100	0.368412	0.891148
5	0.099500	0.386720	0.891148
6	0.071900	0.472839	0.891148
7	0.050400	0.503865	0.897129
8	0.040600	0.521388	0.906699
9	0.034100	0.546760	0.898325
10	0.022800	0.549413	0.901914

Figure 21. Évolution des performances du système.

Afin de comprendre le déroulement des différentes étapes du modèle, on présente dans la section suivante un exemple illustratif détaillé.

- **Exemple d'exécution :**

En utilisant une phrase arabe, nous avons créé une démonstration pratique de la façon avec laquelle notre modèle peut être utilisé. En ce qui concerne, spécifiquement, le traitement des données textuelles dans cette langue particulière, les capacités de notre modèle sont parfaitement mises en évidence.

Tout d'abord, la phrase que nous avons choisie est la suivante :

“ أحب هذا الفيلم كثيرا ! 😊 إنه رائع حقًا. #رائع#سينما قم بزيارة هذا الرائع www.example.com ”

Maintenant, nous allons suivre les étapes de prétraitement (expliquées ci-dessus) et les appliquer à cette phrase :

- ✓ **Suppression des URLs :**

La phrase sera traitée pour supprimer les URLs à l'aide de la fonction **remove_urls**. Après cette étape, la phrase deviendra :

“ أحب هذا الفيلم كثيرا! 😊 إنه رائع حقًا. #رائع#سينما قم بزيارة هذا الموقع الرائع ”

- ✓ **Suppression des caractères spéciaux :**

Ensuite, la phrase sera traitée pour supprimer les caractères spéciaux à l'aide de la fonction **remove_symbols**. Après cette étape, la phrase deviendra :

“ أحب هذا الفيلم كثيرا 😊 إنه رائع حقًا رائع سينم قم بزيارة هذا الموقع الرائع ”

✓ **Suppression des emojis :**

Puis, la phrase on supprime les emojis à l'aide de la fonction **remove_emojis**. La phrase deviendra :

أحب هذا الفيلم كثيرًا إنه رائع حقًا رائع سينما قم بزيارة هذا "

✓ **Segmentation des mots :**

Puis, la phrase est segmentée en mots à l'aide de l'outil **FarasaSegmenter** et elle deviendra :

['أحب', 'هذا', 'الفيلم', 'كثيرًا', 'إنه', 'رائع', 'حقًا', 'رائع', 'سينما', 'قم', 'بزيارة', 'هذا', 'الموقع', 'الرائع']

Ces étapes de prétraitement permettent de nettoyer la phrase en supprimant les URLs, les caractères spéciaux, les emojis, les mots non arabes et de segmenter les mots de manière appropriée.

✓ **Tokenisation :**

Dans cette étape, la phrase sera transformée en codes (tokenisée) à l'aide du tokenizer. Les mots seront convertis en identifiants numériques représentant chaque **token**. Cette phrase sera ensuite ajustée en fonction de la longueur maximale spécifiée et sera prête pour être utilisée dans le modèle. Voici la représentation tokenisée de la phrase :

[2, 229, 173, 82, 2642, 82, 768, 82, 25, 82, 62, 312, 25, 356, 312, 25, 82, 327, 30, 82, 200, 82, 25, 82, 317, 1555, 191, 3]

✓ **Préparation des données :**

Dans cette étape, les données sont préparées en convertissant le format du jeu de données en un format compatible avec le modèle et en utilisant un objet **DataCollatorWithPadding** pour regrouper les données en lots (**batch**) et effectuer le rembourrage (**padding**) lorsque cela est nécessaire.

✓ **Configuration du modèle :**

Une configuration personnalisée est créée en modifiant certains paramètres du modèle Arabert de base, tels que le nombre de couches cachées, le nombre de têtes d'attention, la taille cachée, etc.

✓ **Entraînement du modèle :**

Le modèle est entraîné en utilisant le jeu de données d'entraînement et les paramètres d'entraînement spécifiés. Le modèle ajuste ses poids et biais pour apprendre à prédire les étiquettes de classification.

➤ Evaluation de système ArbPrefRec :

On présente dans cette section, les résultats obtenus lors de l'entraînement de système, en général, et on va les détailler dans le prochain chapitre.

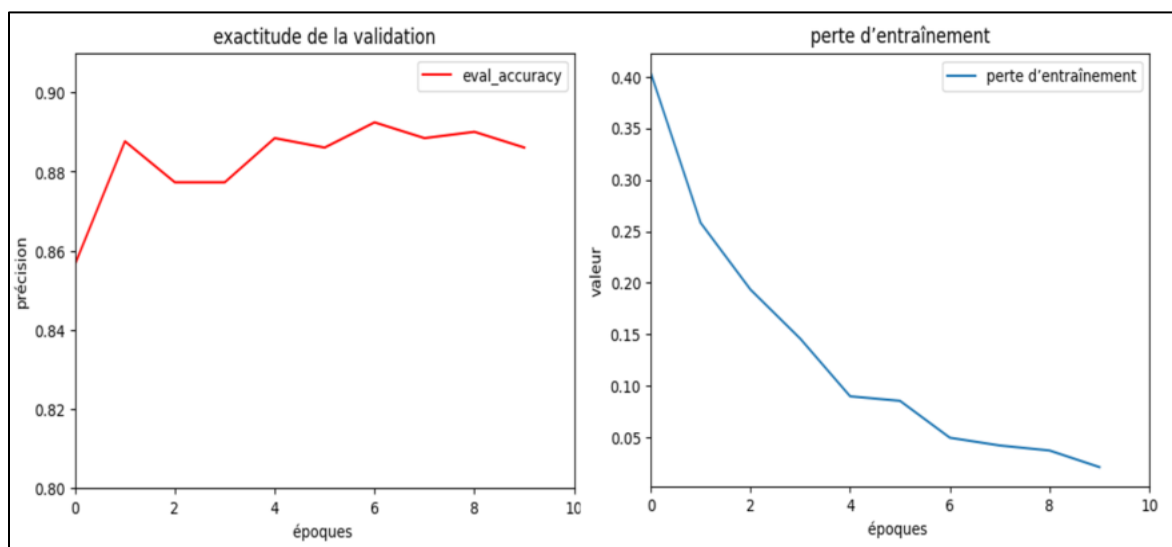


Figure 22. Performance d'entraînement du système ArbPerfRec.

Après avoir entraîné et évalué notre modèle d'analyse des sentiments basé sur **Arabert**, nous avons obtenu des résultats prometteurs en termes de classification précise des commentaires des utilisateurs (positif ou négatif). Cependant, nous avons constaté que ces informations peuvent être encore plus précieuses lorsqu'elles sont utilisées pour fournir une expérience de recommandation personnalisée à nos utilisateurs.

En effet, la phase de formation de notre modèle d'analyse des sentiments nous a permis de comprendre les préférences et les sentiments des utilisateurs concernant différents aspects des produits ou des services. Ces informations constituent une base solide pour formuler des recommandations pertinentes qui répondent aux besoins spécifiques de chaque utilisateur.

Dans les sections suivantes, nous présentons notre approche de recommandation hybride basée sur les préférences et sentiments.

2.6.2. ETAPE 2 : Génération Des Recommandations basées Préférences :

Dans cette section, on présente notre approche de recommandation basée préférences. On propose une nouvelle approche qui combine le **FC** et **FBC**, en incorporant les connaissances acquises grâce à l'analyse des sentiments à partir des commentaires pour identifier les similitudes entre les utilisateurs et recommander des produits ou des services qui ont été appréciés par des utilisateurs ayant des goûts similaires.

En outre, en intégrant des stratégies de recommandation hybrides qui combinent différentes techniques de recommandation, nous pouvons exploiter les forces de chaque approche pour fournir des recommandations plus personnalisées et plus pertinentes, améliorant ainsi la satisfaction et l'engagement de l'utilisateur vis-à-vis du système

En combinant notre modèle d'analyse des sentiments basé sur **Arabert** avec les approches de recommandation citées, on a pu créer un système de recommandation qui utilise les sentiments des utilisateurs pour générer des recommandations personnalisées pertinentes.

Le processus de la recommandation hybride est présenté dans la Figure 2.13.

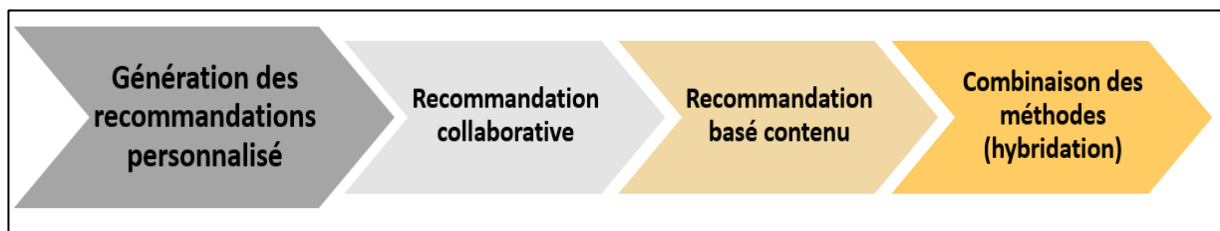


Figure 23. Les étapes de la recommandation hybride.

Dans ce qui suit, nous détaillons les différentes étapes de la recommandation.

a. La Recommandation collaborative :

La recommandation collaborative est une méthode qui identifie les utilisateurs similaires sur la base de leurs préférences et génère des recommandations basées sur les choix des utilisateurs similaires. Pour calculer la similarité, on a utilisé la similarité Cosinus.[51]

$$Similarity(A, B) = \frac{A \cdot B}{|A||B|} \quad (1)$$

Où **A** et **B** sont les vecteurs des évaluations des utilisateurs qui représentent leur préférence ou non de l'item. Autrement dit, c'est les sentiments des utilisateurs envers les items.

Tout d'abord, on commence par la création d'une matrice utilisateur-items avec la fonction **pivot_table()**, où chaque ligne représente un utilisateur et chaque colonne représente un item (restaurant dans notre cas) et les cases contiennent les évaluations des utilisateurs pour chaque item.

restaurant_id	32	33	34	35	36	37	38	39	40	41	...
user_id											
5	1	1	0	0	0	0	0	0	0	0	...
36	0	0	0	0	0	0	0	0	0	0	...
44	0	0	1	0	0	0	0	0	0	0	...
46	0	0	0	1	0	0	0	0	0	0	...
47	0	0	0	0	1	1	0	0	0	0	...
...

Figure 24. Matrice utilisateur-item

Cette matrice nous permet de calculer la similarité entre les utilisateurs et les éléments (i, j) en utilisant la fonction `cosine_similarity()`, cette similarité est calculée entre les évaluations (positives ou négatives) données par les utilisateurs pour les items.

```

1 user_similarity_matrix
array([[1. , 0.2, 0. , ..., 0. , 0. , 0. ],
       [0.2, 1. , 0. , ..., 0. , 0. , 0. ],
       [0. , 0. , 1. , ..., 0. , 0. , 0. ],
       ...,
       [0. , 0. , 0. , ..., 0. , 0. , 0. ],
       [0. , 0. , 0. , ..., 0. , 1. , 0. ],
       [0. , 0. , 0. , ..., 0. , 0. , 0. ]])

```

Figure 25. Matrice de la similarité user-to-user

Ensuite, on prend les K utilisateurs les plus similaires (la valeur de K sera définie expérimentalement dans le chapitre 3). Puis, on vise les items positivement évalués et on va les considérés comme des items candidats pour la recommandation.

La Figure 2.16 présente un exemple de classement des items sur la base des évaluations fournies par les utilisateurs voisins.

Ranked Restaurants based on Labels:	
restaurant_id	
655	1.000000
1238	1.000000
2311	1.000000
2165	1.000000
1827	1.000000
1622	1.000000
4431	1.000000
243	0.928571
1389	0.897959
109	0.888889
33	0.884615
428	0.833333
249	0.785714
357	0.781250
68	0.761194
97	0.733333
990	0.692308
62	0.615385

Name: polarity, dtype: float64

Figure 26. Le classement des items basé sur les sentiments

b. La Recommandation basée sur le contenu :

La recommandation basée sur le contenu est une méthode qui analyse les caractéristiques des items (dans notre cas, les restaurants) pour générer des recommandations similaires.

Nous avons utilisé le contenu des commentaires de l'utilisateur cible sur un item pour calculer la similarité entre son commentaire et ceux des items similaires.

Tout d'abord, les commentaires sont encodés en utilisant le même **tokenizer**. Ensuite, la similarité cosinus est calculée entre les **embeddings** de commentaires correspondants, ce qui permet de mesurer la similarité entre les commentaires de l'utilisateur cible et les commentaires des items.

$$Similarity(A, B) = \frac{A \cdot B}{|A||B|} \quad (2)$$

A. représente le vecteur d'encodage de commentaire de l'utilisateur cible.

B. représente le vecteur d'encodage des commentaires d'un item candidat.

Nous avons ensuite attribué des scores de similarité aux items en fonction de la similarité entre leurs vecteurs de commentaires avec ceux de l'utilisateur cible. Les items sont ensuite classés en fonction de ces scores de similarité et des recommandations sont générées.

Ranked Restaurants based on Comment Scores:		
	restaurant_id	comment_score
2	1689	0.241762
0	732	0.233355
12	237	0.230218
1	1027	0.211412
5	243	0.204343
10	591	0.196124
9	775	0.194409
15	421	0.190073
3	2168	0.185084
11	212	0.184178
13	357	0.178196
7	216	0.163932
14	43	0.160050
8	310	0.160022
6	1175	0.156628
4	204	0.155798

Figure 27. Le classement des items basé sur les commentaires

c. La Recommandation hybride basée sentiments :

Dans notre système de recommandation hybride FC-FBC, on propose d'exploiter, à la fois, les informations basées sur les sentiments (CF) et les informations basées sur les commentaires (CBF) afin de fournir des recommandations précises et personnalisées.

Après avoir classé les items sur la base de leurs évaluations (sentiment) (Figure 2.16) et leur score basé sur les commentaires (Figure 2.17), on a combiné ces deux méthodes de recommandation pour obtenir un classement final des items.

Pour ce faire, on propose une combinaison pondérée du score basé sur les sentiments et de score basé sur les commentaires. Le poids attribué à la note de chaque score détermine son importance relative dans le classement final.

Mathématiquement, la note combinée (Hybrid Score Hs) pour chaque item est calculée comme suit :

$$Hs(u, i) = (1 - \alpha) \times RS(i) + \alpha \times CS(u, i) \quad (3)$$

- **Hs(u,i)**. Représente le score final de l'item (*i*) pour l'utilisateur (*u*).
- **α** le poids attribué pour la combinaison des scores.
- **RS(*i*)** Le score basé sur le sentiment associé à l'item (*i*).
- **CS(*u, i*)** Le score basé sur les commentaires donnés par l'utilisateur (*u*) pour l'item (*i*).

En ajustant le poids α , nous pouvons contrôler l'influence des commentaires sur les recommandations (la valeur de poids sera définie expérimentalement dans le chapitre 3).

Le score basé sur les sentiments représente le classement basé sur la polarité obtenu à partir de notre modèle. Ainsi que Le score basé sur les commentaires reflète la similarité entre les commentaires de l'utilisateur cible et les commentaires associés à chaque item.

En combinant ces deux scores, nous visons à capturer à la fois les signaux de popularité générale capturés par l'approche basée sur les sentiments, et les préférences spécifiques des utilisateurs indiquées par l'approche basée sur les commentaires.

Le paramètre (α) nous permet d'équilibrer l'effet de ces deux scores.

Une fois les scores combinés calculés pour tous les restaurants, ils seront classés par ordre décroissant pour obtenir la liste de recommandations finale.

En intégrant ces deux scores, nous améliorons la précision et la pertinence des recommandations, ce qui se traduit par une amélioration de l'expérience utilisateur.

Plus de détails sur les méthodes de recommandations utilisées dans [l'annexe B](#)

```
↳ Ranked Restaurants based on Combined Scores:
  restaurant_id  combined_score
4              1827      1.000000
1              1238      0.890680
6              4431      0.874513
3              2165      0.859420
5              1622      0.848380
0              655       0.842027
2              2311      0.839932
10             33        0.829465
7              243       0.827044
8              1389      0.812030
11             428       0.793087
9              109       0.786341
12             249       0.725853
14             68        0.685973
13             357       0.679939
15             97        0.672252
16            990       0.633871
17             62        0.577662
```

Figure 28. La Recommandation hybride

2.7. Conclusion :

Dans ce chapitre, nous avons présenté les étapes de conception de notre système de recommandation social basé sur l'élicitation des préférences.

Notre approche proposée, en deux phases, permet la génération des recommandations personnalisées dans un système de recommandation social, en se basant sur les commentaires fournis par les utilisateurs arabophones. En utilisant des techniques d'apprentissage automatique, on a pu comprendre les préférences des utilisateurs et proposer des recommandations adaptées.

Les approches collaboratives et basées sur le contenu ont été utilisées, avec intégration des sentiments et préférences, pour générer des recommandations pertinentes.

Notre solution offre ainsi une expérience utilisateur améliorée dans les systèmes de recommandation sociaux.

Afin d'évaluer la performance de notre système, une série de tests et comparaisons est effectuée et les résultats seront présentés dans le chapitre suivant.

Chapitre 3 : Expérimentations et tests

3.1. Introduction

Dans ce chapitre, nous allons présenter les différents outils et plateformes utilisés pour la mise en œuvre de notre système, ainsi que les métriques d'évaluation utilisées dans nos expériences menées visant à répondre aux questions suivantes :

- ✓ **Question de recherche 1** – Le nouveau modèle AraBERT est-il performant pour l'analyse de sentiment et l'enrichissement du profil utilisateur ?
- ✓ **Question de recherche 2** – Comment réaliser une recommandation pertinente et plaisante à l'utilisateur ?
- ✓ **Question de recherche 3** – Est-ce que les préférences des utilisateurs peuvent aider à améliorer la pertinence des recommandations générées ou non?

3.2. Base de données (DataSet) :

Dans cette section, nous présentons l'ensemble de données 'ar_res_reviews' [W3][47] qui sert de base à notre recherche sur l'analyse des sentiments. Cet ensemble utilisé contient des avis des gens sur les restaurants, qui est dérivé des travaux approfondis menés par **ElSahar** et **El-Beltagy** en (2015) .Leur recherche s'est concentrée sur la création d'une collection complète d'échantillons de textes arabes annotés couvrant divers domaines. [52]

L'ensemble des données comprend un nombre substantiel d'échantillons de textes arabes étiquetés, ce qui constitue une ressource précieuse pour la formation et l'évaluation des modèles d'analyse des sentiments. Les annotations comprennent des étiquettes de sentiments positifs, négatifs et neutres, ce qui nous permet d'explorer l'efficacité de différentes techniques et algorithmes d'apprentissage automatique pour capturer les sentiments exprimés dans le texte arabe.

En tirant parti de cet ensemble de données, nous visons à étudier les performances de notre modèle d'analyse des sentiments et à mieux comprendre les défis et les opportunités associés à l'analyse des sentiments en langue arabe. En outre, cet ensemble de données contient **8364** avis de **3360** utilisateurs de **1650** restaurants d'après **qaym.comt**, il sert de référence pour évaluer la performance de notre modèle par rapport aux approches de pointe existantes en matière d'analyse des sentiments.

3.3. Outils de développements :

3.3.1. Langage de programmation (python) :

Le langage utilisé pour l'implémentation de ce travail est python qui est un langage de programmation de haut niveau axé sur la simplicité et la lisibilité, à jouer un rôle central dans notre étude. Initialement développé par **Guido van Rossum**[\[W6\]](#) et publié pour la première fois en 1991, Python est devenu l'un des langages les plus populaires dans divers domaines, notamment la science des données, l'apprentissage automatique et le traitement du langage naturel.[\[53\]](#)

La simplicité de Python et sa syntaxe claire en font un choix idéal pour les débutants comme pour les programmeurs expérimentés. Sa lisibilité, obtenue grâce à l'utilisation de l'indentation au lieu des crochets, améliore la lisibilité du code et réduit l'encombrement syntaxique. En outre, la vaste bibliothèque standard de Python fournit un ensemble riche de modules et de fonctions à des fins diverses, ce qui simplifie encore les tâches de développement.

L'un des principaux avantages de Python est sa polyvalence et sa compatibilité avec différentes plateformes, notamment Windows, macOS et Linux. En outre, Python peut se targuer d'une communauté dynamique et solidaire, qui favorise le développement de nombreuses bibliothèques et structures répondant à des besoins variés.[\[53\]](#)

En résumé, Python est un langage de programmation polyvalent et largement compatible qui offre un riche ensemble de modules et de fonctions pour diverses tâches de développement. Sa communauté dynamique a développé de nombreuses bibliothèques et frameworks qui répondent à des besoins variés. Python est particulièrement utile dans les applications de science des données et d'apprentissage automatique, et sa simplicité et sa lisibilité en font un choix populaire par rapport à d'autres langages de programmation.

3.3.2. Les bibliothèques utilisées :

Dans le cadre de cette étude, la bibliothèque essentielle utilisée est **Transformers**.

La bibliothèque Transformers, également connue sous le nom de **Hugging Face Transformers**, est une bibliothèque open-source populaire développée par Hugging Face. Elle fournit un moyen simple et efficace d'utiliser et d'affiner des modèles de transformateurs pré-entraînés pour une variété de tâches de traitement du langage naturel (NLP).

Voici quelques caractéristiques et aspects clés de la bibliothèque Transformers :

- ✓ La bibliothèque fournit des modèles de transformateurs pré-entraînés, notamment BERT, GPT, RoBERTa et XLNet, adaptés à des tâches de TAL telles que la classification de textes, la reconnaissance et la génération d'entités.
- ✓ La bibliothèque Transformers prend en charge PyTorch et TensorFlow pour l'apprentissage en profondeur.
- ✓ API de haut niveau : La bibliothèque fournit une API de haut niveau qui simplifie le processus de chargement, d'ajustement et d'évaluation des modèles de transformateurs. Elle fait abstraction de nombreux détails de mise en œuvre et fournit des interfaces faciles à utiliser pour les tâches NLP courantes.
- ✓ La bibliothèque Transformers prend en charge les méthodes de tokenisation telles que WordPiece, BPE et SentencePiece pour la conversion de texte.
- ✓ Ajustement précis (Fine-Tuning) : La bibliothèque permet aux utilisateurs d'affiner les modèles de transformateurs pré-entraînés sur des tâches spécifiques en aval. Elle prend en charge l'apprentissage par transfert, où les modèles sont d'abord pré-entraînés sur des ensembles de données à grande échelle, puis affinés sur des ensembles de données spécifiques à une tâche afin d'améliorer les performances.
- ✓ La bibliothèque Transformers utilise l'architecture des transformateurs, basée sur des mécanismes d'auto-attention, pour une capture efficace des informations contextuelles dans les données textuelles.

Dans l'ensemble, la bibliothèque Transformers s'est imposée comme une ressource incontournable pour de nombreuses tâches NLP en offrant un ensemble complet d'outils et de modèles pré-entraînés qui peuvent être rapidement ajustés et affinés pour répondre aux exigences d'une application particulière.[\[54\]](#),[\[W1\]](#)

3.3.3. Le matériel utilisé :

On a utilisé un pc portable qui a les caractéristiques suivantes :

- Modèle de l'ordinateur : Dell Latitude E7440.
- Mémoire RAM : 8 Go de RAM.
- Stockage : SSD de 256 Go.
- Processeur : Intel Core i7 de 4e génération.
- Architecture : Système d'exploitation 64 bits.

3.4. Les métriques d'évaluation :

Pour évaluer notre système, nous avons utilisé un certain nombre de mesures d'évaluation :

- **Accuracy:** mesure les prédictions correctes par rapport aux prédictions totales en divisant les prédictions correctes par le nombre total d'échantillons.

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (4)$$

TP (True Positives) correspond au nombre de prédictions positives correctes,

TN (True Negatives) correspond au nombre de prédictions négatives correctes,

FP (False Positives) correspond au nombre de prédictions positives incorrectes,

FN (False Negatives) correspond au nombre de prédictions négatives incorrectes.

- **Courbe ROC (receiver operating characteristic) :** La courbe ROC est une représentation graphique des performances d'un modèle de classification à différents seuils de probabilité. Elle représente le taux de vrais positifs par rapport au taux de faux positifs, ce qui permet d'évaluer la sensibilité du modèle à la fois aux vrais et aux faux positifs.
- **Matrice de confusion:** Une matrice de confusion est une représentation tabulaire des prédictions de classification, comparant les prédictions du modèle aux véritables étiquettes de classe. Elle fournit des informations sur les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs.

En utilisant toutes ces métriques, on peut avoir une vue claire sur les performances de notre système.

3.4. Evaluation du système ArbPrefRec et comparaison des résultats :

Dans cette section, on va présenter les résultats d'apprentissage de système en détail et évaluer notre modèle en le comparant avec d'autres modèles qu'on a utilisé sur le même jeu de données. Ensuite, on va présenter l'évaluation de la partie recommandation générée par notre système.

3.4.1. Évaluation du modèle d'analyse de sentiment :

Avant de lancer l'apprentissage de système, on a ajusté plusieurs paramètres et les modifier après chaque entraînement jusqu'à avoir des résultats performants.

Les paramètres optimaux sont présentés dans le tableau 4 :

<i>Paramètre</i>	<i>Valeur</i>
Nombre d'époques d'entraînement	3/10/20
Nombre de couches cachées utilisées	12
Taille du batch par périphérique <i>Batch_size (training/evaluation)</i>	16/32
Taux d'apprentissage (<i>Learning_rate</i>)	1e-5
Coefficient de décroissance du poids (<i>weight_decay</i>)	0.01

Tableau 4. La configuration des paramètres utilisés.

a. Performance du système pour 3 époques :

On a commencé par entraîner le modèle sur 3 époques, les résultats obtenus sont présentés dans la Figure 3.1.

Epoch	Training Loss	Validation Loss	Accuracy
1	0.515000	0.453698	0.788278
2	0.415600	0.426757	0.836124
3	0.355300	0.368877	0.858852

Figure 29. Performance du système en 3 époques.

Les informations indiquées montrent les performances du système au cours de différentes époques. Les mesures d'exactitude augmentent régulièrement à chaque époque, ce qui indique une amélioration des performances du modèle en termes de classification correcte des commentaires et de précision des prédictions des sentiments.

En outre, les valeurs de perte pour l'apprentissage et la validation diminuent au fil des époques, ce qui indique que le système apprend et réduit les erreurs. Les résultats sont présentés dans la Figure 3.2.

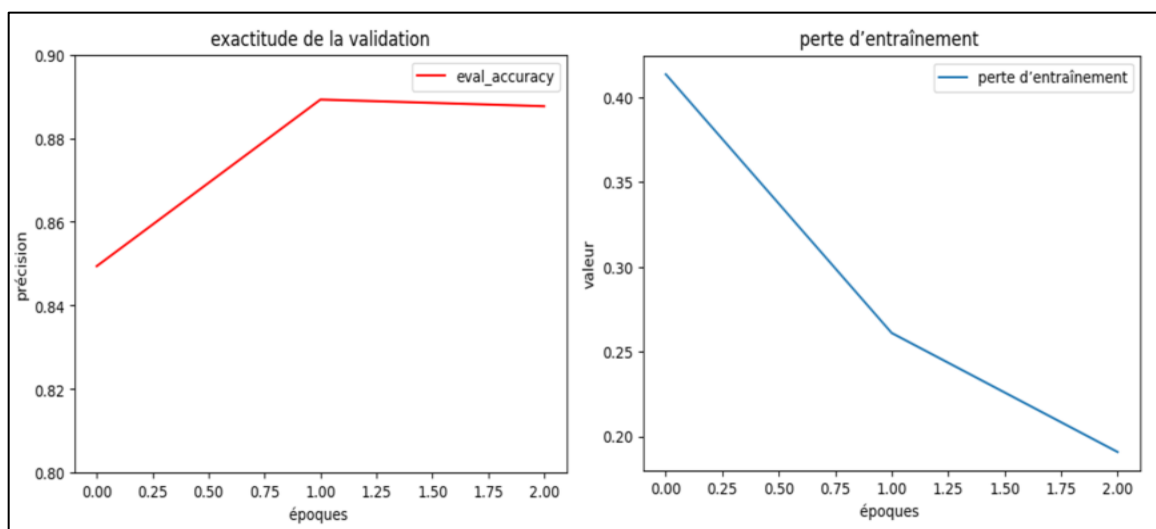


Figure 30. Perte Vs. Précision du système en 3 époques.

La Figure 3.2, montre que le système apprend efficacement à partir des données d'apprentissage et améliore ses performances et se généralise bien aux données non vues.

b. Performance du système pour 10 époques :

En observant le schéma d'amélioration des mesures à travers les époques, il est raisonnable de suggérer que l'augmentation du nombre d'époques pourrait conduire à des meilleurs résultats. Le système semble capable d'apprendre plus à mesure des époques, ce qui suggère qu'une période d'entraînement plus longue pourrait permettre au système de continuer à s'améliorer. Toutefois, il est également important d'être vigilant quant au sur-apprentissage possible, le système risque de trop s'adapter aux données d'apprentissage spécifiques et de ne pas se généraliser de manière optimale à de nouvelles données.

On a donc lancé l'entraînement sur 10 époques, les résultats obtenus sont présentés dans la Figure 3.3.

Epoch	Training Loss	Validation Loss	Accuracy
1	0.403900	0.350963	0.856574
2	0.258400	0.302071	0.887649
3	0.193600	0.409103	0.877291
4	0.145900	0.528498	0.877291
5	0.090000	0.561056	0.888446
6	0.085600	0.612551	0.886056
7	0.049800	0.639725	0.892430
8	0.042300	0.689769	0.888446
9	0.037400	0.717655	0.890040
10	0.021300	0.745778	0.886056

Figure 31. Performance du système en 10 époques.

Les mesures d'évaluation de l'exactitude augmentent globalement, ce qui montre une amélioration des performances de système. Toutefois, il est important de surveiller attentivement la relation entre la perte de validation et ces mesures afin d'évaluer la capacité du modèle à se généraliser correctement. Les résultats de performance du système sont présentés dans la Figure 3.4.

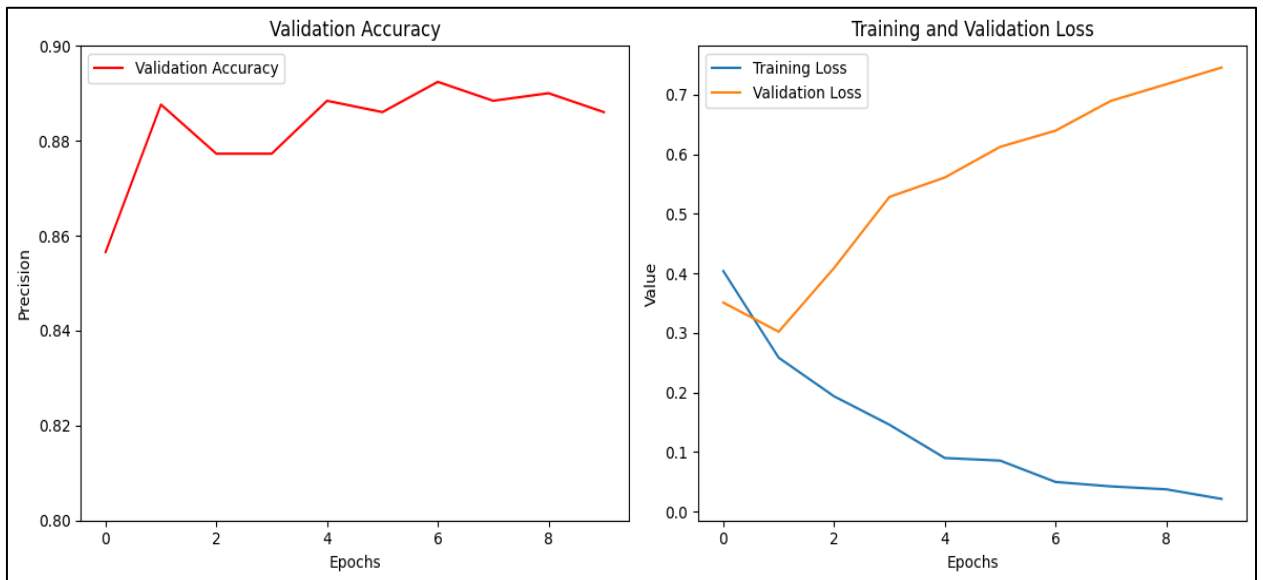


Figure 32. Précision Vs. Perte en 10 époques

Les résultats présentés dans la Figure 3.4, montrent que la perte d'apprentissage diminue progressivement au fil des époques, ce qui indique que le système s'adapte aux données d'apprentissage. Cependant, la perte de validation présente un schéma plus complexe, avec des augmentations observées à certaines époques, ce qui peut indiquer des difficultés à généraliser le système.

c. Performance du système dans 20 époques :

Les résultats de performance du système pour 20 époques sont présentés dans la Figure 3.5.

Epoch	Training Loss	Validation Loss	Accuracy
1	0.351500	0.269969	0.896414
2	0.258800	0.260384	0.897211
3	0.170600	0.373463	0.899602
4	0.117000	0.471684	0.892430
5	0.073200	0.525142	0.895618
6	0.079600	0.592629	0.899602
7	0.053400	0.561983	0.894821
8	0.035000	0.679699	0.889243
9	0.015100	0.755760	0.891633
10	0.017100	0.741302	0.901992
11	0.008600	0.819106	0.899602
12	0.008800	0.867136	0.894821
13	0.017800	0.885985	0.891633
14	0.005400	0.917813	0.889243
15	0.021200	0.877596	0.898805
16	0.004600	0.887553	0.898008
17	0.002600	0.878911	0.898805
18	0.004800	0.893903	0.900398
19	0.011900	0.921123	0.897211
20	0.002700	0.910723	0.899602

Figure 33. Performance du système en 20 époques.

D'après la Figure 3.5, on voit que les performances du système s'améliorent au fil du temps. La perte d'apprentissage diminue progressivement, ce qui indique que le système s'adapte aux données d'apprentissage. Cependant, la perte de validation montre une certaine variation, ce qui peut indiquer que le système à rencontrer des difficultés.

Les différents résultats sont schématisés dans la Figure 3.6.

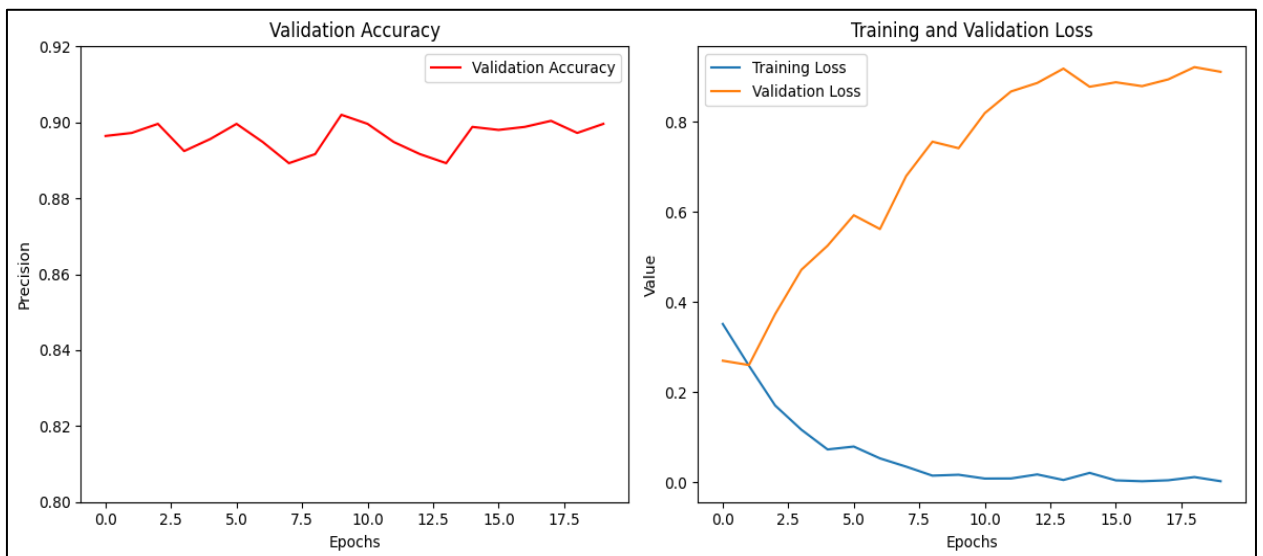


Figure 34. Précision Vs. Perte en 20 époques.

- **Discussion des résultats obtenus :**

En comparant les résultats obtenus dans les trois premières époques avec ceux des dix et vingt époques, on constate une amélioration significative des mesures d'exactitude. Cela suggère que le système a continué à apprendre et à s'améliorer au cours des époques supplémentaires.

Cependant, il est important de noter que la perte de validation a augmenté pour quelques époques supplémentaires, ce qui peut indiquer le début d'un sur-apprentissage. Cela souligne l'importance de surveiller attentivement les mesures de validation et d'appliquer des techniques de régularisation pour éviter le sur-apprentissage et améliorer la capacité du modèle à se généraliser correctement.

En conclusion, l'entraînement de système sur un plus grand nombre d'époques a permis d'améliorer les performances globales, mais il est essentiel de tenir compte de la perte de validation lors de l'évaluation de la capacité du modèle à se généraliser correctement.

Après avoir utilisé le système pour la prédiction, on présente les résultats avec la courbe Roc et matrice de confusion. Les résultats sont présentés dans la Figure 3.7.

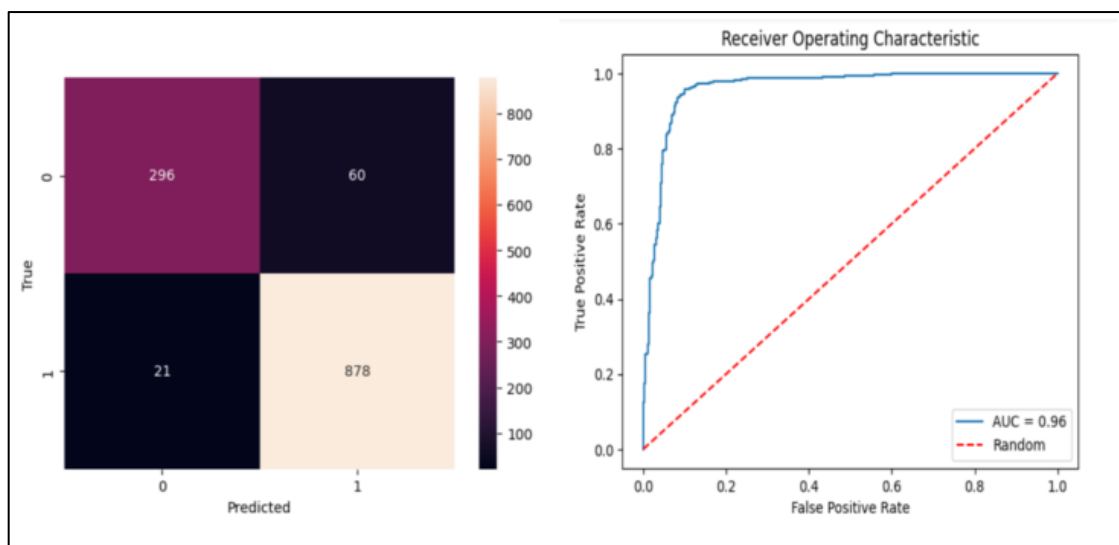


Figure 35. La Courbe ROC et matrice de confusion.

A partir de la Figure 3.7, la courbe ROC montre le compromis entre le taux de vrais positifs et le taux de vrais négatifs à différents seuils. Dans notre cas, la courbe ROC indique une excellente performance avec une précision de 0,96. Une valeur de précision plus élevée indique un meilleur rendement du modèle. Cela signifie que le modèle réussit à classer correctement les commentaires positives tout en maintenant un taux de faux positifs relativement faible.

3.4.2. Comparaison des résultats :

Afin de prouver la performance de notre modèle, on a choisi deux modèles du deep learning, les plus célèbres dans le domaine du traitement automatique de langage naturel et de la classification.

Le premier modèle est le modèle **LSTM** (*Long Short-Term Memory*) basé sur une couche, pour capturer les dépendances séquentielles dans les données. La couche d'embedding convertit les entiers en vecteurs denses, puis la couche **LSTM** traite la séquence d'entrée en conservant la mémoire à long terme. Enfin, une couche dense avec une activation **softmax** est utilisée pour générer les probabilités de classe.

Le modèle est compilé avec l'optimiseur **Adam** et utilise une perte "**categorical_crossentropy**" pour mesurer la différence entre les prédictions et les étiquettes réelles, la précision est utilisée comme métrique pour évaluer les performances du modèle.

Le deuxième modèle est le modèle **CNN** (*Convolutional Neural Network*) qui utilise une architecture basée sur des couches convolutionnelles pour l'apprentissage des caractéristiques. Il commence par une couche **d'embedding** qui convertit les entiers en vecteurs denses. Ensuite, une couche **Conv1D** applique des filtres convolutifs unidimensionnels pour extraire des caractéristiques locales. La couche **GlobalMaxPooling1D** réduit la dimensionnalité en prenant les valeurs maximales de chaque canal. Ensuite, une couche dense effectue une transformation linéaire suivie d'une activation **ReLU** pour introduire de la non-linéarité.

Une couche **Dropout** désactive aléatoirement certains neurones pour éviter le sur-apprentissage. Enfin, la couche de sortie utilise une activation **sigmoïde** pour la classification binaire.

Avant de comparer les résultats obtenus par notre modèle avec ces deux modèles, on va d'abord présenter brièvement leurs résultats d'apprentissage et de test. Ensuite, on va les comparer avec ceux de notre système.

Nous présentons dans ce qui suit les résultats de chaque modèle séparément.

- **Modèle 1 : Modèle LSTM**

Les résultats obtenus avec une Architecture composée d'une couche d'intégration, suivie d'une couche LSTM de 128 unités et d'une couche dense avec activation SoftMax avec 20 époques sont présentés dans la Figure 3.8.

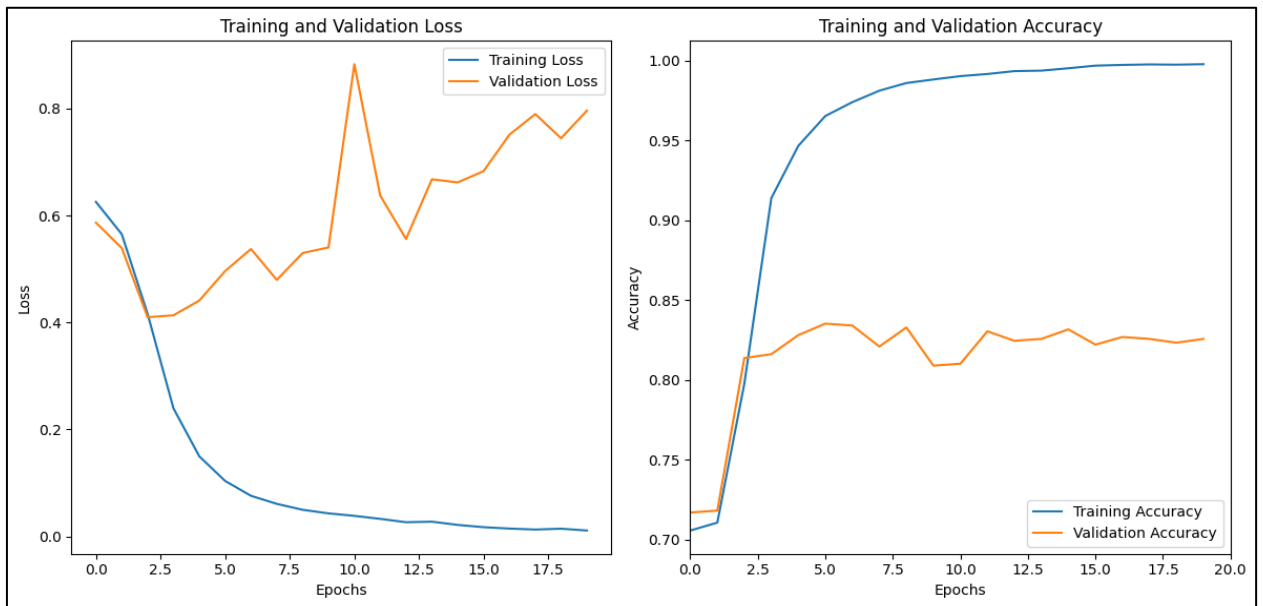


Figure 36. Performance du modèle LSTM.

D’après la Figure 3.8, après l’entraînement du modèle pendant 20 époques, on observe une diminution de la perte d’entraînement et une augmentation de la précision. Cependant, les résultats de validation ne suivent pas la même tendance. La perte de validation augmente légèrement tandis que la précision stagne autour de 0,82. Cela suggère que le modèle peut être en train de trop ajuster les données d’entraînement et ne se généralise pas bien sur de nouvelles données.

- **Modèle 2 :** Le modèle CNN

Les résultats de performance du modèle CNN avec une architecture composée d’une couche d’intégration, suivie d’une couche convolutive 1D avec 128 filtres et une taille de noyau de 5, d’une couche de mise en commun du maximum global et de deux couches denses avec activation relu, Le modèle est entraîné à l’aide de l’optimiseur d’Adam avec une perte d’entropie croisée binaire, et entraîné pendant 10 époques avec une taille de lot de 32 sont présentés dans la figure 3.9.

Notons que les deux modèles LSTM et CNN utilisent une couche d’intégration pour représenter les mots dans un espace vectoriel continu.

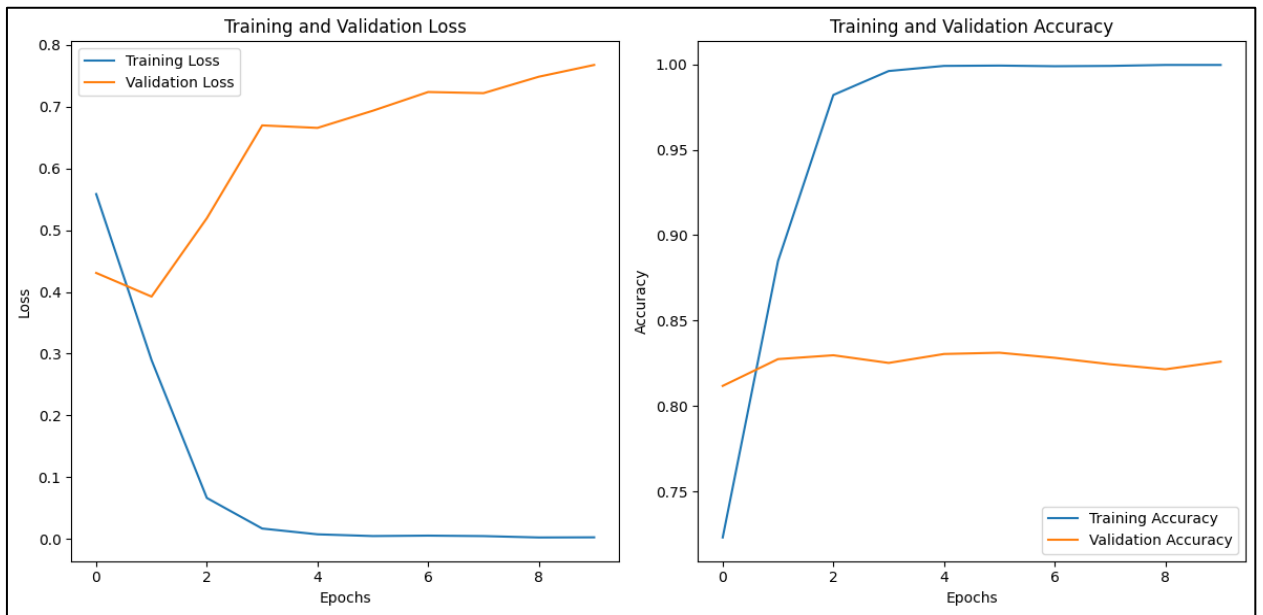


Figure 37. Performance du modèle CNN.

A partir de la Figure 3.9 et après 10 époques d'entraînement, on observe que le modèle a montré de bonnes performances d'entraînement avec une perte décroissante et une précision croissante. Cependant, les résultats de validation ont été moins satisfaisants, avec une légère augmentation de la perte de validation et une précision de validation stagnante autour de 0,82.

- **Comparaison entre les trois modèles :**

La Figure 3.10 présente les résultats de performance des trois modèles afin d'avoir une vue claire sur la différence entre eux et d'exprimer une comparaison entre eux.

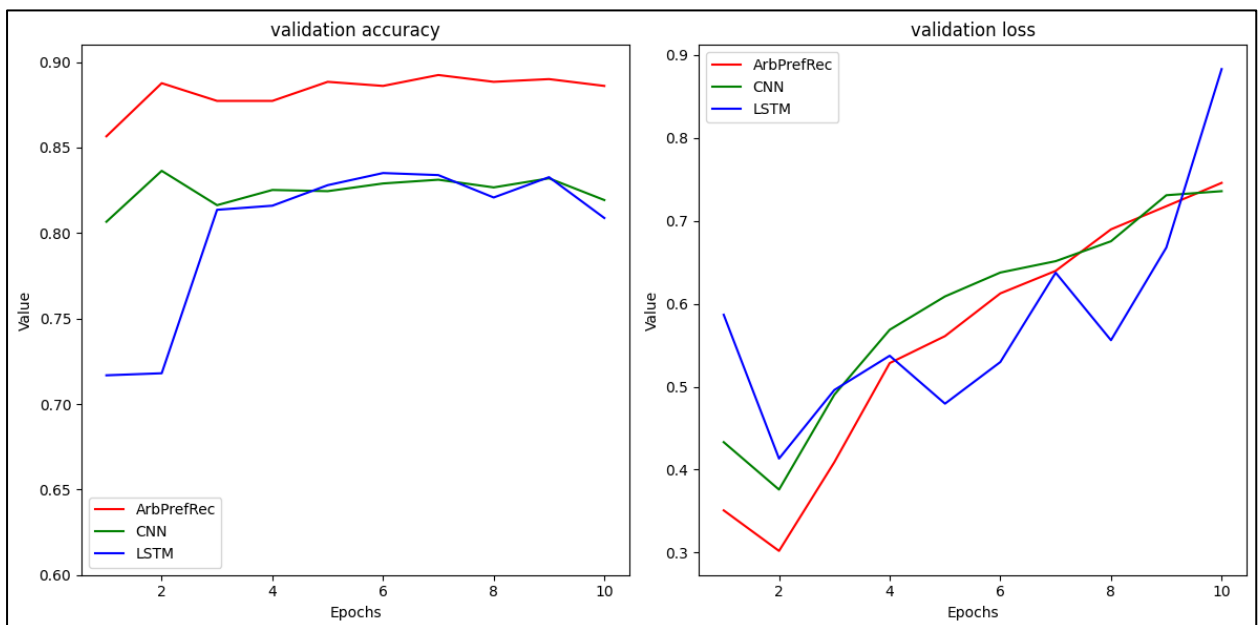


Figure 38. Comparaison entre les trois modèles.

Les valeurs de comparaison de la précision d'entraînement et du test entre les trois modèles sont présentées dans le tableau 5.

Modèle	LSTM	CNN	ArbPrefRec
Train Accuracy	0.993	0.999	0.901
Test Accuracy	0.622	0.832	0.956

Tableau 5. Comparaison entre les trois modèles sur la partie de test.

En analysant les résultats de performance des trois modèles, nous pouvons observer que le modèle LSTM atteint une précision d'apprentissage élevée de 0,993, mais qu'il ne se généralise pas bien aux données de test non vues, ce qui se traduit par une précision de test plus faible de 0,622. Tandis que le modèle CNN est encore plus performant en termes de précision d'apprentissage, atteignant 0,999, et montre une meilleure généralisation avec une précision de test plus élevée de 0,832.

Cependant, le système ArbPrefRec, malgré une précision d'apprentissage inférieure de 0,901 par rapport aux modèles LSTM et CNN, il les surpasse tous les deux en termes de précision de test avec un score impressionnant de 0,956. Cela indique que notre système est plus performant que les modèles LSTM et CNN en termes de prédiction.

Le système ArbPrefRec se caractérise par sa taille importante par rapport aux modèles LSTM et CNN. Sa taille est mesurée en fonction du nombre de paramètres, qui déterminent la complexité et la capacité du modèle à apprendre des relations dans les données. Cette complexité permet au système de capturer des informations plus fines et des relations plus complexes entre les mots d'un texte arabe.

Par conséquent, sur la base des résultats obtenus, on peut conclure que notre modèle est plus performant en termes de précision globale sur les données de test, surpassant à la fois les deux modèles, LSTM et CNN.

On passe maintenant à l'évaluation de l'approche de recommandation proposée.

3.5. Evaluation de la recommandation hybride basée préférences:

Avant d'évaluer les recommandations générées, tous d'abord il faut expliquer le choix des valeurs optimales des 3 paramètres utilisés dans la recommandation :

- Le paramètre K (nombre des utilisateurs similaires),
- Le seuil qu'on doit utiliser pour sélectionner les restaurants candidats en se basant sur les sentiments,

- Le poids α (utilisé pour la combinaison des recommandations dans schème hybride).

3.5.1. Réglage des paramètres :

Afin de trouver les valeurs optimales des différents paramètres, on a choisi un intervalle pour chacun d'eux et on a analysé les résultats pour les différentes combinaisons possibles avec

$K = [5, 10, 15]$; seuil = $[0.7, 0.8, 0.9]$; et $\alpha = [0.2, 0.35, 0.5]$.

On note que durant tout ce qui suit, on va considérer que les recommandations qui ont un score total supérieur à 0.90 comme des recommandations pertinentes.

Le résultat obtenu pour l'utilisateur actif ayant 'user_id' = 5 sont présentés dans la Figure 3.11.

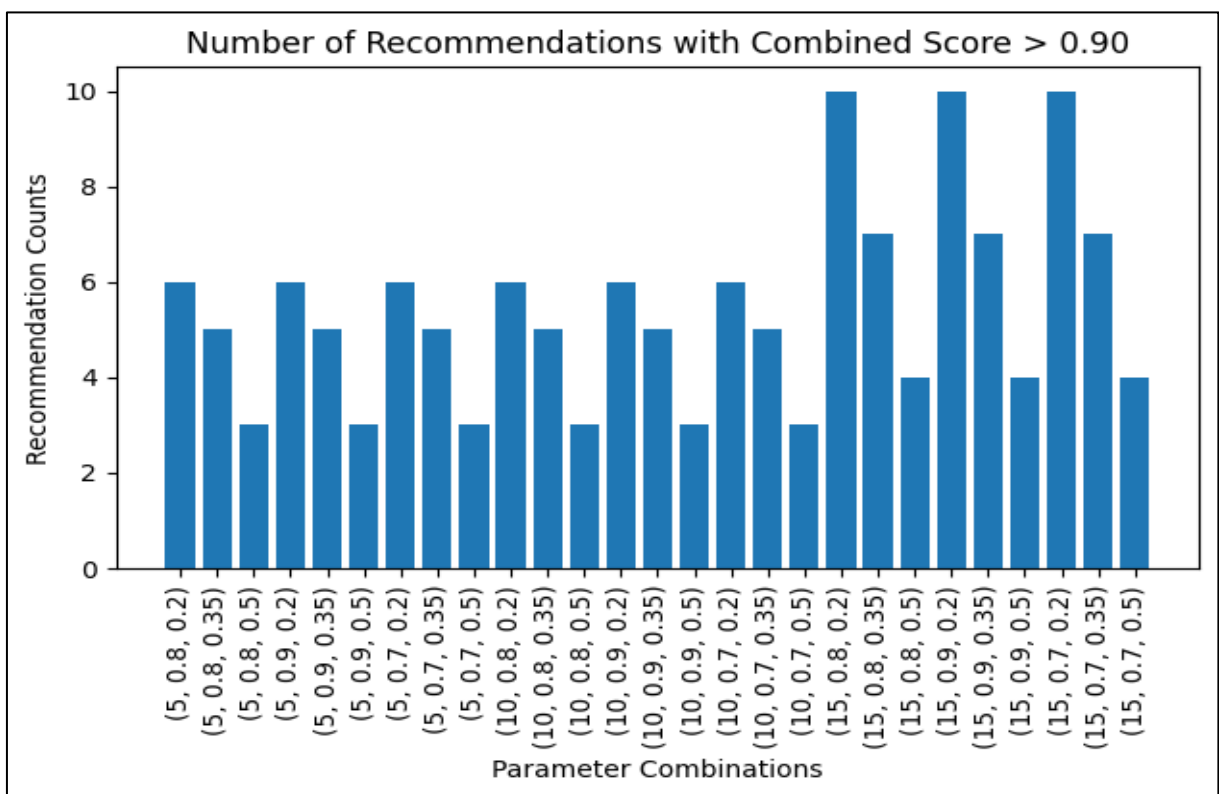


Figure 39. l'effet des valeurs de paramètre sur le nombre de recommandation.

A partir des résultats présentés dans la Figure 3.11, on va discuter les différentes valeurs optimales des paramètres.

- ✓ **Paramètre K (nombre de voisins) :** Les combinaisons de paramètres avec $K = 5$ et $K = 10$ donnent systématiquement des nombres de recommandations de 6.

La combinaison de paramètres avec $K = 15$ montre une amélioration du nombre de recommandations avec un nombre de 10, ce qui indique que l'augmentation du nombre de voisins peut conduire à de meilleurs résultats.

- ✓ **Seuil :** Les combinaisons de paramètres avec un seuil de 0.7, 0.8, 0.9 ne montrent aucun changement des résultats avec les différents cas ce qui signifie sa faible importance, donc on a choisi la valeur 0.8.
- ✓ **Poids de fusion :** Les combinaisons de paramètres avec un poids de fusion de 0,2 produisent systématiquement des nombres de recommandations plus élevés, avec des nombres de 6, 7 et 10 pour différentes valeurs de K et de seuil. Lorsque la pondération des commentaires augmente jusqu'à 0,35 et 0,5, le nombre de recommandations diminue, passant de 3 à 4.

Sur la base de cette analyse, nous pouvons observer ce qui suit :

- ✓ Le maintien de $K = 15$ conduit régulièrement à des recommandations plus nombreuses.
- ✓ L'utilisation d'une pondération de 0,2 pour les commentaires (FBC) tend à produire un plus grand nombre de recommandations.

Ces observations suggèrent que des valeurs plus élevées pour K et une pondération des commentaires de 0,2 peuvent être considérées comme des choix de paramètres favorables pour optimiser le nombre de recommandations pertinentes.

3.5.2. Évaluation des performances du système de recommandation ArbPrefRec :

Dans le but de tester la capacité du système à donner des recommandations pertinentes, on le lance pour un nombre d'utilisateur égal à (10, 20, 30, 40, 50,100) respectivement. A chaque fois on calcule sa capacité à générer au moins 10 recommandations, avec un score total supérieur à 0.90 (seuil de pertinence). On note que les utilisateurs ont été sélectionnés aléatoirement.

Les résultats sont présentés dans la Figure 3.12 :

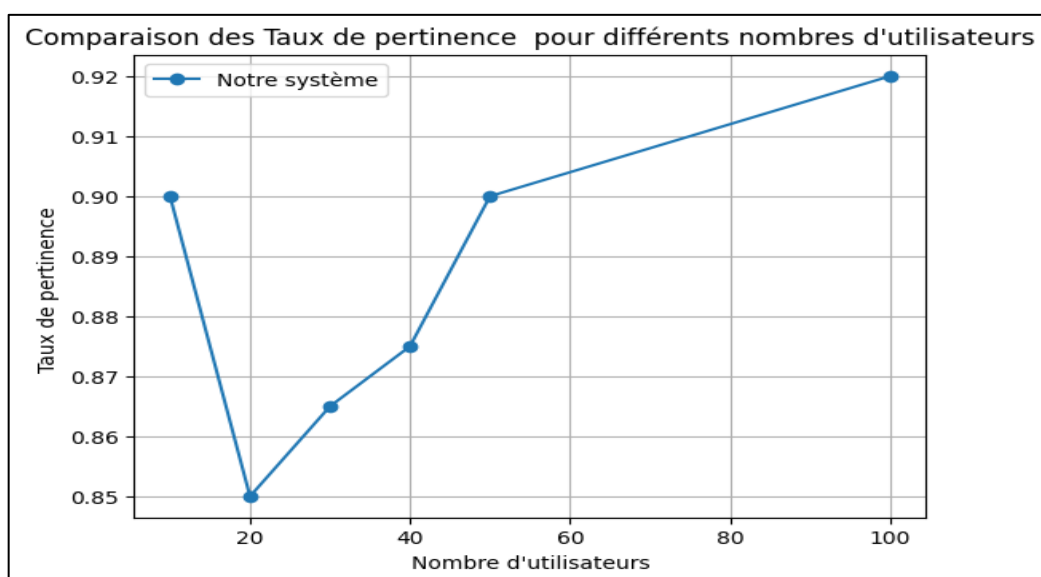


Figure 40. Taux de pertinence pour différents nombres d'utilisateurs.

En analysant les résultats présentés dans la Figure 3.12, on distingue plusieurs observations intéressantes :

- *Augmentation du Taux de pertinence* : Nous constatons une légère tendance à l'augmentation du taux de réussite en fonction du nombre d'utilisateurs. Par exemple, le taux de réussite passe de 0,9 pour 10 et 20 utilisateurs à 0,92 pour 100 utilisateurs. Cela peut être attribué à la disponibilité de plus de données sur les utilisateurs, ce qui permet au système de faire des recommandations plus précises et d'améliorer les résultats.

En résumé, ces résultats montrent que le système est performant avec des taux de réussite cohérents pour différents nombres d'utilisateurs. Le système présente un potentiel d'évolutivité et montre une corrélation positive entre le nombre d'utilisateurs et le taux de réussite.

3.5.3. Comparaison des résultats de recommandation :

Afin de tester d'avantage notre système, et vu le manque des ressources de données disponibles dans notre jeu des données utilisé, on a trouvé des difficultés à appliquer les métriques d'évaluation à cause d'absence des données de recommandation correctes (manque d'information concernant les recommandations attendues pour chaque utilisateur) afin de calculer la précision et le rappel.

Pour cela, on a proposé une nouvelle méthode pour l'évaluation du système. La méthode consiste à générer des recommandations pour différents utilisateurs sur la base des prédictions de notre système et les comparer avec les recommandations basées sur le modèle LSTM.

On va comparer les résultats trouvés pour les deux types de recommandations Top-10 et Top-20.

- **La recommandation Top-10 :**

La Figure 3.13 présente les résultats de performances des deux systèmes recommandations, ArabPrefRec et basé LSTM, respectivement, en sélectionnant les TOP 10 restaurants.

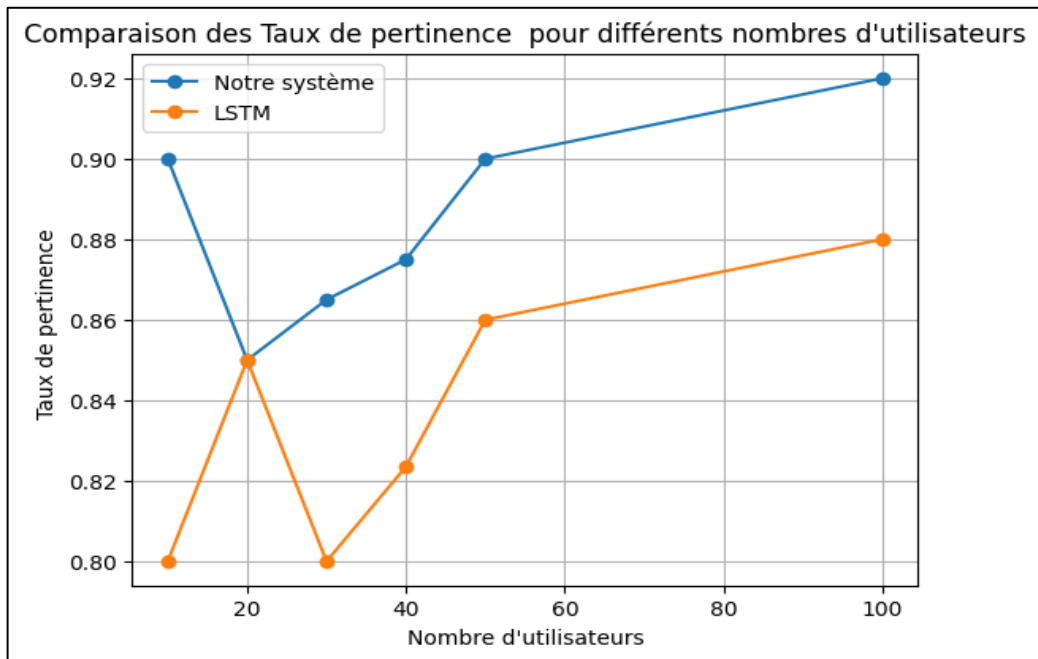


Figure 41. Comparaison des résultats de recommandation pour TOP-10.

Ces résultats montrent que notre système a généralement obtenu des taux de réussite supérieurs à ceux de l'autre système pour chaque nombre d'utilisateurs évalué. Cela suggère que notre système présente une meilleure performance en termes de recommandations personnalisées pour un nombre croissant d'utilisateurs.

- **La recommandation Top-20 :**

La Figure 3.14 présente les résultats de performances des deux systèmes recommandations, ArabPrefRec et basé LSTM, respectivement, en sélectionnant les TOP 20 restaurants.

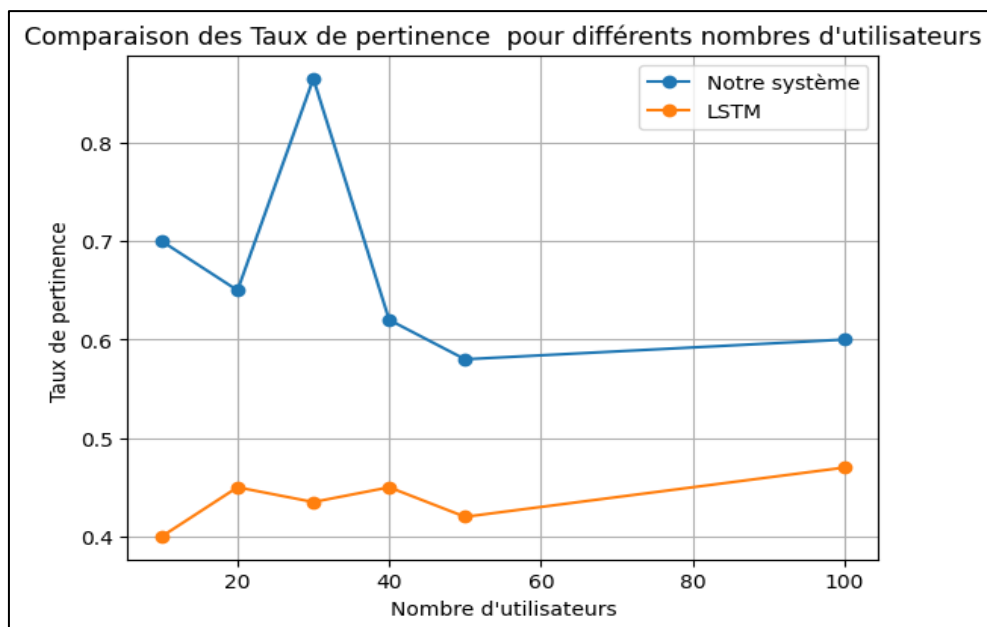


Figure 42. Comparaison des résultats de recommandation pour Top-20.

A partir de la Figure 3.14, on peut observer que notre système de recommandation présente des taux de pertinence plus élevés que le système basé sur LSTM pour tous les nombres d'utilisateurs analysés.

Pour notre système, les taux de pertinence varient entre 0,7 et 0,87 pour des nombres d'utilisateurs allant de 10 à 30. Cependant, ces taux de réussite diminuent pour des nombres d'utilisateurs supérieurs à 30, atteignant 0,58 pour 50 utilisateurs et 0,6 pour 100 utilisateurs.

En revanche, pour le système basé LSTM, les taux de réussite sont globalement plus faibles, variant entre 0,4 et 0,46 pour les mêmes nombres d'utilisateurs.

Dans l'ensemble, notre système de recommandation semble avoir des taux de réussite plus élevés que le système basé LSTM dans le cas des deux recommandations Top-10 et Top-20 pour différents nombres d'utilisateurs.

Donc, on peut conclure que le système **ArabPrefRec** est plus performant que le système basé LSTM en termes de pertinence des recommandations ce qui revient à la performance du modèle d'élicitation des préférences utilisé.

Après avoir présenté les différents tests et expérimentations conduits afin de prouver la performance de notre système de recommandation hybride basé sur l'élicitation des préférences, nous présentons dans la section suivante son interface.

3.6. Présentation de l'interface du système de recommandation ArbPrefRec :

On présente dans cette section l'interface de notre système de recommandation des restaurants qui utilise permet aux utilisateurs de découvrir les différents restaurants disponibles et de fournir leurs avis (commentaire) en arabe.

L'interface du système ArbPrefRec offre une visualisation du fonctionnement du système de recommandation et permet aux utilisateurs d'obtenir des recommandations personnalisées et de prédire les résultats en fonction de leurs commentaires. Cette partie présente quelques informations sur l'interface du système ArbPrefRec.

L'interface est conçue de manière conviviale, offrant une expérience utilisateur intuitive. Les utilisateurs peuvent facilement naviguer à travers les différentes fonctionnalités du système, notamment la recherche de recommandations, la soumission de commentaires et l'affichage des résultats de la recommandation.

La Figure 3.15 présente l'interface principale de notre système **ArabPrefRec**.

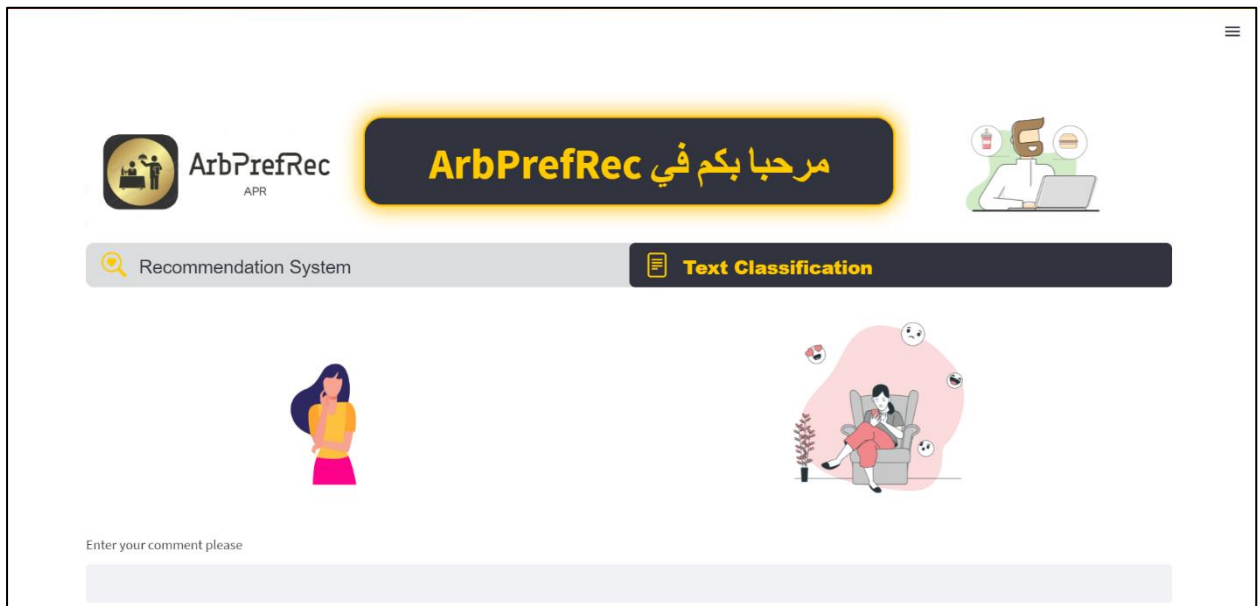


Figure 43. L'interface de système ArbPrefRec.

Lorsque les utilisateurs accèdent à l'interface, ils sont accueillis par une page d'accueil attrayante qui présente les fonctionnalités principales du système. Une barre de navigation permet aux utilisateurs de spécifier leurs préférences et d'obtenir des recommandations personnalisées en fonction de leurs intérêts.

Une fois que les utilisateurs ont obtenu leurs recommandations, ils peuvent explorer les différents résultats suggérés et d'une note de popularité. Cela permet aux utilisateurs de prendre des décisions claires sur les restaurants qui les intéressent.

Les résultats de recommandation pour l'utilisateur actif ayant User-ID= 5 sont présentés dans la Figure 3.16.

Recommendation for User ID: 5

Positive Restaurants by Target User:

	restaurant_id	name_rest
0	1181	Tavern Eatery

	Similar Users:
0	19105
1	34703
2	81
3	31663
4	30245
5	7707
6	587
7	3812
8	2267
9	444

	Ranked Restaurants :	name restaurant :	score :
0	1690	Cafe Cuisine	1.0000
1	1181	Tavern Eatery	0.9998
2	64	Cafe Spot	0.9988
3	3954	Bistro Place	0.9963
4	607	Bistro Restaurant	0.9961
5	52	Diner House	0.9921
6	136	Bistro Eatery	0.9914
7	1127	Cafe Eatery	0.9849
8	51	Diner Cuisine	0.9830
9	1019	Bistro Spot	0.9818

Figure 44. Affichage des résultats de recommandations.

De plus, l'interface permet également aux utilisateurs de soumettre leurs propres commentaires sur les restaurants recommandés. Ces commentaires sont utilisés pour améliorer les prédictions du système et affiner les recommandations futures comme il est présenté dans la Figure 3.17.

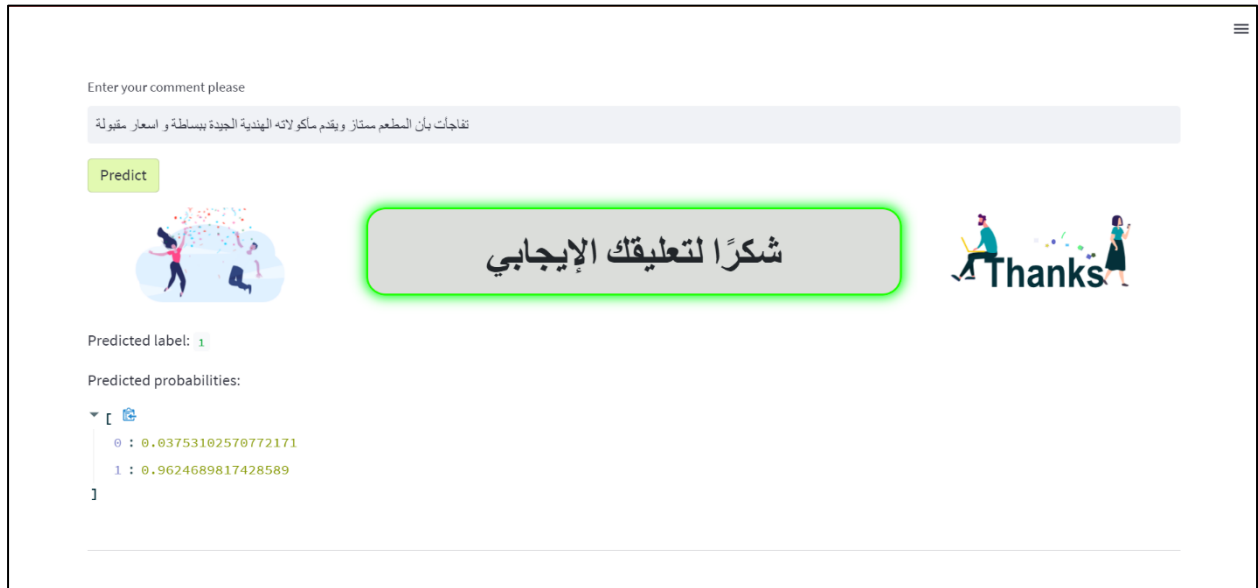


Figure 45. Résultats de prédiction de système ArbPrefRec sur les commentaires.

A part de la Figure 3.16, on voit que l'interface affiche également des statistiques sur les performances du système, telles que la précision des recommandations et la satisfaction des utilisateurs. Ces informations fournissent aux utilisateurs une rétroaction sur la qualité du système et permettent aux développeurs de prendre des mesures pour améliorer continuellement l'expérience utilisateur.

En résumé, l'interface du système ArbPrefRec offre aux utilisateurs une expérience conviviale pour accéder aux recommandations personnalisées et prédire les résultats en fonction de leurs commentaires écrits en arabe. Elle facilite la navigation et l'interaction avec le système, permettant ainsi aux utilisateurs de tirer pleinement parti des fonctionnalités offertes par le système de recommandation.

3.7. Conclusion

Dans ce chapitre, nous avons présenté les différents outils et bibliothèques utilisés pour l'implémentation de notre système. De plus, nous avons réalisé une série de tests pour évaluer les performances de notre système. Puis, nous avons effectué une comparaison entre notre système et les modèles les plus utilisés, sachant LSTM et CNN, pour évaluer le modèle de l'élicitation des préférences ainsi que l'approche de recommandation hybride proposée.

On a constaté que l'utilisation du modèle **AraBERT** est plus bénéfique et elle peut apporter un plus à la recherche surtout dans le domaine du TALN et elle peut remplacer d'autres modèles comme LSTM et CNN.

De plus, les résultats obtenus ont été globalement satisfaisants, et nous ont permis de mettre en évidence les avantages et les limites de chaque approche. Ces résultats ont ouvert la voie à de nouvelles perspectives de recherche, qui seront abordées dans la conclusion.

Conclusion générale et Perspectives

Le travail présenté dans ce mémoire se situe dans le domaine du Traitement du Langage Naturel, plus précisément, l'élicitation des préférences à partir de commentaires dans les Systèmes de Recommandation Sociaux. Notre étude se concentre sur l'analyse des sentiments, en utilisant le modèle **AraBERT** développé récemment par Google pour améliorer les requêtes, la prédiction et la génération de textes rédigés en Arabe, en plus de l'utilisation de ces sentiments pour l'amélioration de la qualité des recommandations générées.

Notre objectif était l'exploitation des caractéristiques et des points forts de ce modèle afin de l'utiliser dans l'élicitation des préférences à partir des commentaires écrits en arabe et de réaliser des recommandations pertinentes grâce à une approche de recommandation hybride.

Pour mieux comprendre notre travail, nous avons d'abord présenté de manière approfondie le domaine de la recommandation sociale et de l'élicitation des préférences, en mettant l'accent sur l'analyse des sentiments. Ensuite, nous avons introduit le modèle **AraBERT** en présentant les travaux existants dans l'état de l'art.

Par la suite, nous avons décrit notre approche qui consistait à mettre en place une solution pour l'analyse des sentiments (positifs/négatifs) à partir des commentaires fournis par les utilisateurs arabophones. Pour cela, nous avons utilisé l'algorithme d'apprentissage profond AraBERT pour extraire les préférences (sentiments) à partir du texte saisi par les utilisateurs.

Puis, les sentiments et préférences prédites ont été employés dans une nouvelle approche de recommandation hybride. Afin d'améliorer la qualité des recommandations générées et d'augmenter la performance du système, l'approche proposée combine un FC basé sentiments avec un Filtrage par Contenu basé sur les préférences extraites depuis les commentaires.

Ensuite, une série de tests et d'expérimentations est menée pour illustrer les différentes propositions et prouver la performance du système. En ce qui concerne les résultats de l'analyse des sentiments basée sur AraBERT, les performances de classification étaient relativement acceptables et ont montré une plus grande performance par rapport aux résultats obtenus avec les modèles LSTM et CNN. De plus, La recommandation hybride basée préférence en utilisant le modèle AraBert était plus précise et plus pertinente par rapport à celle générée en utilisant le modèle LSTM, ce qui prouve l'efficacité de nos propositions et répondent à nos objectifs visés.

En termes de perspectives futures, plusieurs axes de recherche peuvent être envisagés. Tout d'abord, il serait intéressant d'explorer les capacités du système en l'adaptant à la langue

dialectale algérienne « Darija ». De plus, il serait très important de penser de relier le système avec d'autres méthodes issues la recommandation inter-domaines (Cross domaine recommandation) pour traiter le problème du démarrage à froid dans les systèmes de recommandation. Une dernière perspective pour ce travail est son application sur d'autres Datasets de taille plus grande afin de tester sa performance réelle.

A. Références Bibliographiques

- [1] J. Devlin, M.-W. Chang, K. Lee, et K. Toutanova, « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ». *arXiv*, 24 mai 2019. Consulté le: 21 mai 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/1810.04805>
- [2] W. Antoun, F. Baly, et H. Hajj, « AraBERT: Transformer-based Model for Arabic Language Understanding ». *arXiv*, 7 mars 2021. Consulté le: 21 mai 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/2003.00104>
- [3] Guy, Ido, and David Carmel. "Social recommender systems." *In Proceedings of the 20th international conference on companion on World wide web*, pp. 283-284, 2011.
- [4] J. Tang, X. Hu, et H. Liu, « Social recommendation: a review », *Soc. Netw. Anal. Min.*, vol. 3, n° 4, p. 1113-1133, 2013, doi: 10.1007/s13278-013-0141-9.
- [5] J. Shokeen et C. Rana, « Social recommender systems: techniques, domains, metrics, datasets and future scope », *J. Intell. Inf. Syst.*, vol. 54, n° 3, p. 633-667, juin 2020, doi: 10.1007/s10844-019-00578-5.
- [6] J. J. Pazos Arias, A. Fernández Vilas, et R. P. Díaz Redondo, *Recommender Systems for the Social Web*, vol. 32. in *Intelligent Systems Reference Library*, vol. 32. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-25694-3.
- [7] J. Shokeen et C. Rana, « A study on features of social recommender systems », *Artif. Intell. Rev.*, vol. 53, n° 2, p. 965-988, févr. 2020, doi: 10.1007/s10462-019-09684-w.
- [8] N. Zheng et Q. Li, « A recommender system based on tag and time information for social tagging systems », *Expert Syst. Appl.*, vol. 38, n° 4, p. 4575-4587, avr. 2011, doi: 10.1016/j.eswa.2010.09.131.
- [9] J. Shokeen, and R. Chhavi. "A review on the dynamics of social recommender systems." *International Journal of Web Engineering and Technology* . 2018.
- [10] Y. Koren, « Collaborative filtering with temporal dynamics », *Commun. ACM*, vol. 53, n° 4, p. 89-97, avr. 2010, doi: 10.1145/1721654.1721677.
- [11] I. Fernández-Tobías, et al. "Cross-domain recommender systems: A survey of the state of the art." *In Proceedings of the Spanish conference on information retrieval*. Vol. 24, 2012.

- [12] X. Yang, Y. Guo, Y. Liu, et H. Steck, « A survey of collaborative filtering based social recommender systems », *Comput. Commun.*, vol. 41, p. 1-10, mars 2014, doi: 10.1016/j.comcom.2013.06.009.
- [13] E. Çano et M. Morisio, « Hybrid Recommender Systems: A Systematic Literature Review », *Intell. Data Anal.*, vol. 21, n° 6, p. 1487-1524, nov. 2017, doi: 10.3233/IDA-163209.
- [14] M. C. B. Natividad, B. D. Gerardo, et R. P. Medina, « A fuzzy-based career recommender system for senior high school students in K to 12 education », *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 482, p. 012025, mars 2019, doi: 10.1088/1757-899X/482/1/012025.
- [15] C. Guan, Y. Kevin Kam Fung, et Y. Yue, « Towards a Personalized Item Recommendation Approach in Social Tagging Systems Using Intuitionistic Fuzzy DBSCAN », in *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou: IEEE, août 2018, p. 361-364. doi: 10.1109/IHMSC.2018.00090.
- [16] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, et Y.-C. Zhang, « Solving the apparent diversity-accuracy dilemma of recommender systems », *Proc. Natl. Acad. Sci.*, vol. 107, n° 10, p. 4511-4515, mars 2010, doi: 10.1073/pnas.1000488107.
- [17] X. N. Lam, , et T. Vu "Addressing cold-start problem in recommendation systems." *In Proc. of the 2nd Inter. Conf. on Ubiquitous information management and communication*. 2008.
- [18] J. Bobadilla, F. Ortega, A. Hernando, et A. Gutiérrez, « Recommender systems survey », *Knowl.-Based Syst.*, vol. 46, p. 109-132, juill. 2013, doi: 10.1016/j.knosys.2013.03.012.
- [19] N. Idrissi et A. Zellou, « A systematic literature review of sparsity issues in recommender systems », *Soc. Netw. Anal. Min.*, vol. 10, n° 1, p. 15, 2020, doi: 10.1007/s13278-020-0626-2.
- [20] J. O'Donovan et B. Smyth, « Is trust robust?: an analysis of trust-based recommendation », in *Proceedings of the 11th international conference on Intelligent user interfaces*, Sydney Australia: ACM, janv. 2006, p. 101-108. doi: 10.1145/1111449.1111476.
- [21] S. K. "Tony" Lam, D. Frankowski, et J. Riedl, « Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems », in *Emerging Trends in Information and Communication Security*, G. Müller, Éd., in LNCS, Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 3995, p. 14-29, 2006. doi: 10.1007/11766155_2.
- [22] U. Gretzel et D. R. Fesenmaier, « Persuasiveness of Preference Elicitation Processes in Destination Recommendation Systems », in *Information and Communication Technologies in Tourism 2005*, A. J. Frew, Éd., Vienna: Springer-Verlag, 2005, p. 194-204. doi: 10.1007/3-211-27283-6_18.

- [23] R. Katarya et O. P. Verma, « An effective web page recommender system with fuzzy c-mean clustering », *Multimed. Tools Appl.*, vol. 76, n° 20, p. 21481-21496, oct. 2017, doi: 10.1007/s11042-016-4078-7.
- [24] S. Deng, L. Huang, G. Xu, X. Wu, et Z. Wu, « On Deep Learning for Trust-Aware Recommendations in Social Networks », *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, n° 5, p. 1164-1177, mai 2017, doi: 10.1109/TNNLS.2016.2514368.
- [25] J. Capdevila, M. Arias, et A. Arratia, « GeoSRS: A hybrid social recommender system for geolocated data », *Inf. Syst.*, vol. 57, p. 111-128, avr. 2016, doi: 10.1016/j.is.2015.10.003.
- [26] S. Ahmadian, M. Meghdadi, et M. Afsharchi, « A social recommendation method based on an adaptive neighbor selection mechanism », *Inf. Process. Manag.*, vol. 54, n° 4, p. 707-725, juill. 2018, doi: 10.1016/j.ipm.2017.03.002.
- [27] Y. Shen, et al. "A collaborative filtering based social recommender system for E-commerce." *International Journal of Simulation: Systems, Science and Technology*, vol 17, n° 22 pp:91-96, (2016). doi: 10.5013/IJSSST.a.17.22.09.
- [28] L. Chen, and P. Pu. "Survey of preference elicitation methods." *No. REP_WORK*. 2004.
- [29] B. Duthil, "De l'extraction des connaissances à la recommandation." *Diss. Ecole Nationale Supérieure des Mines d'Alès*, 2012.
- [30] P. Pu, L. Chen, et R. Hu, « Evaluating recommender systems from the user's perspective: survey of the state of the art », *User Model. User-Adapt. Interact.*, vol. 22, n° 4-5, p. 317-355, oct. 2012, doi: 10.1007/s11257-011-9115-7.
- [31] H. Yang, S. Sanner, G. Wu, et J. P. Zhou, « Bayesian Preference Elicitation with Keyphrase-Item Coembeddings for Interactive Recommendation », in *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, Utrecht Netherlands: ACM, juin 2021, p. 55-64. doi: 10.1145/3450613.3456814.
- [32] H. ABBACI, « Elicitation des Préférences dans les Systèmes de Recommandation Conversationnels ». Diss 30 juin 2022.
- [33] L. Yue, W. Chen, X. Li, W. Zuo, et M. Yin, « A survey of sentiment analysis in social media », *Knowl. Inf. Syst.*, vol. 60, n° 2, p. 617-663, août 2019, doi: 10.1007/s10115-018-1236-4.
- [34] L. Zhang, S. Wang, et B. Liu, « Deep learning for sentiment analysis: A survey », *WIREs Data Min. Knowl. Discov.*, vol. 8, n° 4, juill. 2018, doi: 10.1002/widm.1253.

- [35] G. Li et Q. Chen, « Exploiting Explicit and Implicit Feedback for Personalized Ranking », *Math. Probl. Eng.*, vol. 2016, p. 1-11, 2016, doi: 10.1155/2016/2535329.
- [36] H. Sobhanam et A. K. Mariappan, « Addressing cold start problem in recommender systems using association rules and clustering technique », in *2013 International Conference on Computer Communication and Informatics*, Coimbatore, Tamil Nadu, India: IEEE, janv. 2013, p. 1-5. doi: 10.1109/ICCCI.2013.6466121.
- [37] I. Elnabarawy, W. Jiang, et D. C. Wunsch II, « Survey of Privacy-Preserving Collaborative Filtering ». arXiv, 18 mars 2020. Consulté le: 21 mai 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/2003.08343>
- [38] O. Sar Shalom, S. Berkovsky, R. Ronen, E. Ziklik, et A. Amihod, « Data Quality Matters in Recommender Systems », in *Proceedings of the 9th ACM Conference on Recommender Systems*, Vienna Austria: ACM, sept. 2015, p. 257-260. doi: 10.1145/2792838.2799670.
- [39] M. Kunaver et T. Požrl, « Diversity in recommender systems – A survey », *Knowl.-Based Syst.*, vol. 123, p. 154-162, mai 2017, doi: 10.1016/j.knosys.2017.02.009.
- [40] F. Ben Kharrat, A. Elkhleifi, et R. Faiz, « Recommendation system based contextual analysis of Facebook comment », in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, Agadir, Morocco: IEEE, nov. 2016, p. 1-6. doi: 10.1109/AICCSA.2016.7945792.
- [41] W. Paik, S. Yilmazel, E. Brown, M. Poulin, S. Dubon, et C. Amice, « Applying natural language processing (NLP) based metadata extraction to automatically acquire user preferences », in *Proceedings of the 1st international conference on Knowledge capture*, Victoria British Columbia Canada: ACM, , p. 116-122, oct. 2001. doi: 10.1145/500737.500757.
- [42] R. Khezzar, A. Moursi, et Z. Al Aghbari, « arHateDetector: detection of hate speech from standard and dialectal Arabic Tweets », *Discov. Internet Things*, vol. 3, n° 1, p. 1, mars 2023, doi: 10.1007/s43926-023-00030-9.
- [43] S. Shahriar, N. Al Roken, et I. Zualkernan, « Classification of Arabic Poetry Emotions Using Deep Learning », *Computers*, vol. 12, n° 5, p. 89, avr. 2023, doi: 10.3390/computers12050089.
- [44] A. Wadhawan, « AraBERT and Farasa Segmentation Based Approach For Sarcasm and Sentiment Detection in Arabic Tweets ». arXiv, 2 mars 2021. Consulté le: 31 mai 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/2103.01679>

- [45] S. Aftan et H. Shah, « Using the AraBERT Model for Customer Satisfaction Classification of Telecom Sectors in Saudi Arabia », *Brain Sci.*, vol. 13, n° 1, p. 147, janv. 2023, doi: 10.3390/brainsci13010147.
- [46] A. Al-Tamimi, A. Shatnawi, and E. Bani-Issa. "Arabic sentiment analysis of YouTube comments." *In the IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), IEEE*, p. 1-6, 2017.
- [47] T. H. Soliman, M. A. Elmasry, A. Hedar, et M. M. Doss, « Sentiment Analysis of Arabic Slang Comments on Facebook », *Int. J. Comput. Technol.*, vol. 12, n° 5, p. 3470-3478, janv. 2014, doi: 10.24297/ijct.v12i5.2917.
- [48] H. El Moubtahij, H. Abdelali, et E. B. Tazi, « AraBERT transformer model for Arabic comments and reviews analysis », *IAES Int. J. Artif. Intell. IJ-AI*, vol. 11, n° 1, p. 379, mars 2022, doi: 10.11591/ijai.v11.i1.pp379-387.
- [49] A. Abdelali, K. Darwish, N. Durrani, et H. Mubarak, « Farasa: A Fast and Furious Segmenter for Arabic », in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, San Diego, California: Association for Computational Linguistics, 2016, p. 11-16. doi: 10.18653/v1/N16-3003.
- [50] A. Vaswani, et al. "Attention is all you need." *In Proc of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, (2017).
- [51] B. Sarwar, G. Karypis, J. Konstan, et J. Riedl, « Item-based collaborative filtering recommendation algorithms », in *Proceedings of the 10th international conference on World Wide Web*, Hong Kong Hong Kong: ACM, avr. 2001, p. 285-295. doi: 10.1145/371920.372071.
- [52] A. Gelbukh et A. Gelbukh, "Computational linguistics and intelligent text processing", In the Proceedings of the 16th International Conference on CICLing, Cairo, Egypt, 2015. in Lecture notes in computer science, no. 9041. Cham: Springer, 2015.
- [53] M. Nosrati, « Python: An appropriate language for real world programming », *World Appl. Program.*, vol. 1, n° 2, p. 110-117, 2011.
- [54] M. Pavlovic, T. Heinis, F. Tauheed, P. Karras, et A. Ailamaki, « TRANSFORMERS: Robust spatial joins on non-uniform data distributions », in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, Helsinki, Finland: IEEE, mai 2016, p. 673-684. doi: 10.1109/ICDE.2016.7498280.

B. Références Web (Techniques)

- [W1] **huggingface** Transformerspage (consulté le 07 Mai, 2023),
<https://huggingface.co/docs/transformers/index/>
- [W2] <http://jalammar.github.io/illustrated-bert/>
- [W3] Jeu de donnée ‘ar_res_reviews’
https://huggingface.co/datasets/ar_res_reviews
- [W4] <https://huggingface.co/aubmindlab/bert-base-arabert>
- [W5] Recall, Precision, F1 Score – Explication Simple Métrique en ML,
<https://inside-machinelearning.com/recall-precision-f1-score/>
- [W6] « History and License », *Python documentation*.
<https://docs.python.org/3/license.html>
(consulté le 20 avr 2023).

Le modèle **Arabert** est une architecture de réseau neuronal basée sur les transformateurs, spécialement conçue pour le traitement du langage naturel en arabe. Il s'agit d'une variante du modèle BERT (Bidirectional Encoder Representations from Transformers) qui a été pré-entraînée sur un vaste corpus de données en langue arabe.

L'architecture du modèle Arabert repose sur des mécanismes de transformer qui permettent de capturer les relations contextuelles et les dépendances entre les mots d'une phrase. Il utilise des techniques avancées telles que l'attention multi-tête, le masquage de tokens et l'apprentissage de séquences à séquences pour extraire des représentations sémantiques riches et précises du langage arabe.

Le modèle Arabert a été évalué dans diverses tâches de traitement du langage, notamment la classification de texte, la génération de texte et la réponse aux questions. Il a démontré d'excellentes performances, rivalisant voire surpassant les autres modèles disponibles pour la langue arabe.

Dans notre étude, nous exploitons les capacités du modèle Arabert pour l'élicitation des préférences et la génération de recommandations dans les systèmes de recommandation sociale. En utilisant Arabert, nous sommes en mesure d'analyser les commentaires des utilisateurs, d'extraire les sentiments et les préférences exprimés, et de générer des recommandations personnalisées basées sur ces informations.

Grâce à son entraînement sur des données en langue arabe, le modèle Arabert est capable de comprendre les subtilités linguistiques spécifiques à l'arabe, telles que les variations morphologiques et les structures grammaticales complexes. Cela lui confère un avantage significatif dans la compréhension et l'analyse des textes arabes, ce qui améliore la qualité et la pertinence des recommandations proposées aux utilisateurs arabophones.

En conclusion, le modèle Arabert représente une avancée majeure dans le domaine du traitement automatique du langage en arabe. Son utilisation dans les systèmes de recommandation sociale ouvre de nouvelles perspectives pour une personnalisation plus précise des recommandations et une meilleure satisfaction des utilisateurs arabophones.

La recommandation hybride est une approche qui combine la recommandation collaborative et l'approche basée sur le contenu pour améliorer la précision et la pertinence des recommandations. Dans cette annexe, nous présentons une méthode spécifique de recommandation hybride qui intègre ces deux approches.

La recommandation collaborative se base sur les préférences des utilisateurs similaires pour générer des recommandations. Elle repose sur le principe que les utilisateurs partageant des goûts similaires ont tendance à apprécier des éléments similaires. Cependant, cette approche peut être limitée lorsque les données sur les préférences des utilisateurs sont rares ou peu fiables.

Pour surmonter ces limitations, nous intégrons également une approche basée sur le contenu. Cette approche analyse les caractéristiques des éléments recommandés, tels que leur description, leurs tags ou leurs attributs, pour proposer des recommandations en se basant sur la similarité entre ces caractéristiques. Ainsi, même en l'absence de données sur les préférences des utilisateurs, nous pouvons recommander des éléments pertinents en se basant sur leurs caractéristiques intrinsèques.

L'approche hybride que nous présentons dans cette annexe combine la recommandation collaborative avec l'approche basée sur le contenu de manière synergique. Elle utilise à la fois les informations sur les utilisateurs similaires et les caractéristiques des éléments recommandés pour générer des recommandations plus personnalisées et pertinentes. En intégrant les avantages de ces deux approches, la recommandation hybride permet de fournir des recommandations plus précises, adaptées aux préférences individuelles des utilisateurs.

En conclusion, la recommandation hybride qui combine la recommandation collaborative avec l'approche basée sur le contenu est une approche puissante pour améliorer la qualité des recommandations dans les systèmes de recommandation. Elle permet de tirer parti des informations sur les utilisateurs similaires tout en utilisant les caractéristiques des éléments recommandés, offrant ainsi une expérience de recommandation plus personnalisée et satisfaisante pour les utilisateurs.

Résumé

Ce mémoire se focalise sur l'élicitation des préférences dans les Systèmes de Recommandation Sociaux en utilisant des techniques du Traitement Automatique du Langage Naturel. L'étude inclut une analyse approfondie du domaine de la recommandation sociale et de l'élicitation des préférences, en mettant l'accent sur l'analyse des sentiments. Pour effectuer cette dernière, on a utilisé le modèle **AraBERT** récemment développé par Google, qui améliore le traitement des requêtes, la prédiction et la génération de texte en Arabe. Notre objectif est d'exploiter les avantages d'**AraBERT** pour l'élicitation des préférences à partir des commentaires en arabe et l'incorporation de ces préférences dans le processus de la recommandation afin d'améliorer les performances des systèmes de recommandations sociaux en termes de pertinence.

Un système de recommandation social des restaurants nommé '**ArbPrefRec**' basé sur les préférences extraites depuis les commentaires des utilisateurs arabophones a été implémenté avec le langage de programmation python avec une interface conviviale.

Une série de test a été effectuée afin de démontrer la performance du système. En comparons leur performance avec les modèles **LSTM** et **CNN**. Les résultats trouver en était satisfaisant du cote classification des sentiments et les recommandations génères.

Mots clés : système de recommandation social, élicitation des préférences, analyse des sentiments, recommandation hybrides, AraBert.

Abstract

This thesis focuses on preference elicitation in Social Recommendation Systems using Natural Language Processing techniques. The study includes an in-depth analysis of the domain of social recommendation and preference elicitation, with a focus on sentiment analysis. To carry out the latter, we used the **AraBERT** model recently developed by Google, which improves query processing, prediction and text generation in Arabic. Our aim is to exploit the advantages of **AraBERT** for eliciting preferences from Arabic comments and incorporating these preferences into the recommendation process in order to improve the performance of social recommendation systems in terms of relevance.

A social restaurant recommendation system called '**ArbPrefRec**' based on preferences extracted from Arabic user comments was implemented using the Python programming language with a user-friendly interface.

A series of tests was carried out to demonstrate the performance of the system. We compared its performance with that of the **LSTM** and **CNN** models. The results were satisfactory in terms of the classification of sentiments and the recommendations generated.

Keywords: social recommendation system, preference elicitation, sentiment analysis, hybrid recommendation, AraBert.

ملخص

تركز هذه الأطروحة على استنباط الأفضلية في أنظمة التوصيات الاجتماعية باستخدام تقنيات معالجة اللغة الطبيعية. تتضمن الدراسة تحليلاً متعمقاً لمجال التوصية الاجتماعية واستئارة الأفضليات، مع التركيز على تحليل المشاعر. لتنفيذ هذا الأخير، استخدمنا نموذج **AraBERT** الذي طوره Google مؤخراً، والذي يحسن معالجة الاستفسارات والتنبؤ وتوليد النصوص باللغة العربية. هدفنا هو استغلال مزايا **AraBERT** للحصول على أفضليات من التعليقات العربية وإدراج هذه الأفضليات في عملية التوصية من أجل تحسين أداء أنظمة التوصيات الاجتماعية من حيث الأهمية.

تم تنفيذ نظام توصيات المطاعم الاجتماعية المسمى '**ArbPrefRec**' بناءً على التفضيلات المستخلصة من تعليقات المستخدم العربي باستخدام لغة برمجة Python مع واجهة سهلة الاستخدام.

وأجريت سلسلة من الاختبارات لإثبات أداء النظام. قارنا أداؤها بأداء طرازي **LSTM** و **CNN**. وكانت النتائج مرضية من حيث تصنيف المشاعر والتوصيات الصادرة.

الكلمات الرئيسية: نظام التوصية الاجتماعية، استنباط الأفضليات، تحليل المشاعر، التوصية الهجينة، **AraBert**.