

# Génération séquentiel prenant en compte des informations contextuelles en nutrition

## Système de recommandation

Alexandre Combeau<sup>1,2</sup>, Vincent Guigue<sup>2</sup>, Cristina Manfredotti<sup>2</sup>, Fatiha Saïs<sup>1</sup>,  
Stephane Dervaux<sup>2</sup>, Paolo Viappiani<sup>3</sup>

<sup>1</sup> AgroParisTech, INRAE, UMR

<sup>2</sup> Université Paris-Saclay

<sup>3</sup> Université Paris Dauphine - PSL

15 mars 2025

### Résumé

Dans le domaine de la nutrition, des recommandations personnalisées sont essentielles pour encourager des habitudes alimentaires saines et adaptées. Cet article présente un système de recommandation exploitant des données séquentielles structurées ainsi que des informations contextuelles afin de proposer des repas adaptés. Notre modèle, basé sur les Réseaux Neuronaux Récurrents, est spécialement conçu pour fournir des recommandations tenant compte des profils nutritionnels individuels, de la diversité des repas et des contraintes de santé. Notre système est également capable de générer des repas complets structurés respectant les contraintes nutritionnelles du monde réel, démontrant ainsi son applicabilité pratique. En plus de surpasser de manière significative les systèmes de pointe sur cette tâche, nous montrons que notre approche séquentielle exploite efficacement la modélisation du contexte.

### Mots-clés

Système de recommandation, Génération de séquences, Modélisation du contexte, Réseau de Neurones Récurrent, Sciences de la santé

## 1 Introduction

La plupart des maladies chroniques telles que le diabète, l'obésité et les maladies cardiovasculaires sont corrélées à des habitudes alimentaires peu saines [26]. Les agences de santé publique ont élaboré des recommandations nutritionnelles destinées à la population afin d'encourager de meilleures habitudes alimentaires. Ce sont les recommandations très médiatisés, par exemple « mangez au moins 5 fruits ou légumes par jour », « limitez votre consommation de sel » L'adhésion du grand public à ces recommandations reste relativement faible, bien que la sensibilisation aux régimes alimentaires sains soit plutôt bonne [19]. Différents facteurs contribuent à cette situation : préférences culturelles et personnelles, difficulté de mise en œuvre, disponibilité et prix des aliments [37], etc. Les systèmes de recommandation (RS) peuvent jouer un

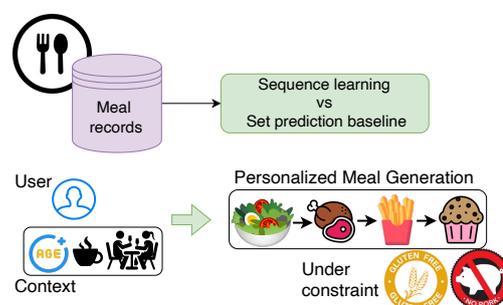


FIGURE 1 – Génération de séquences personnalisées sous contrainte

rôle essentiel pour répondre à ce problème, en proposant des suggestions concrètes et personnalisées tout en respectant des contraintes de diversité, de santé, d'éthique et d'économie. Le défi technique est de concevoir un système capable de recommander non seulement des aliments correspondant aux préférences des utilisateurs, mais aussi des menus, voire des séquences de menus, respectant des contraintes spécifiques (e.g., régimes particuliers, allergies, limites d'apport énergétique...).

Les systèmes de recommandation ont considérablement évolué au cours des 20 dernières années : tandis que les méthodes classiques adoptent des techniques telles que les algorithmes basés sur le contenu [32] et le filtrage collaboratif [25], les systèmes modernes exploitent des architectures d'apprentissage profond capables de prendre en compte différents types de contenu. [11].

Pour évaluer l'efficacité des algorithmes de recommandation dans des tâches qui vont au-delà des contextes traditionnels de prédiction de classement, plusieurs métriques d'évaluation ont été définies [34] et il a été suggéré de prendre en compte les séquences d'actions des utilisateurs pendant leur session [14, 15]. Les domaines de recherche actuels incluent la diversité des recommandations, l'explication associée à celles-ci [31], le contexte général dans le-

quel elle est faite [2] et la transparence du système ou son équité [33].

Alors que la recommandation de recettes est un problème classique, la recommandation de menus complets ouvre de nouvelles perspectives et défis techniques.

Afin de recommander un ensemble cohérent d'aliments sur plusieurs menus, il est nécessaire d'introduire des contraintes diététiques pertinentes [9, 12]. Les travaux précédents [14] se sont concentrés sur l'utilisation des données séquentielles. L'objectif de cet article est de construire une séquence d'aliments constituant un menu tout en intégrant les préférences de l'utilisateur et en maintenant la cohérence du menu (*par exemple*, éviter de proposer du chou-fleur en dessert). De plus, le défi est également de prendre en compte le contexte de l'utilisateur (*par exemple*, les saisons, manger à la maison ou au bureau) [2], ainsi que des contraintes strictes en termes de choix (*par exemple* végétalien, végétarien), religions ou allergies.

Notre proposition consiste à développer un système de recommandation (RS) capable non seulement de compléter une séquence d'aliments, comme dans la génération de playlists [8], mais aussi de générer des séquences complètes sous plusieurs contraintes [18]. À notre connaissance, il n'existe pas de RS capables de générer des séquences complètes et cohérentes (*i.e.*, des menus, qui sont des listes ordonnées d'articles alimentaires) de manière personnalisée et en prenant en compte des contraintes.

En particulier, ce travail va au-delà des propositions précédentes à l'intersection de l'apprentissage automatique et de la nutrition, qui ont présenté la substituabilité des aliments [3] ou la recommandation personnalisée [13]. Notre objectif est de développer un système qui pourrait être utilisé par les nutritionnistes pour trouver des menus sur mesure et répondre à certaines contraintes nutritionnelles.

Dans ce travail, nous exploitons le jeu de données INCA2 [4], qui est un rapport contenant des logs détaillés et contextualisés de consommation alimentaire. Nous décrivons par la suite, notre approche qui utilise une architecture de réseau de neurones récurrent (RNN) pour apprendre le modèle qui générera des séquences alimentaires, comme illustré dans la Fig. 1.

Afin de démontrer l'importance de considérer l'aspect séquentiel de notre problème, nous comparons la performance de notre approche basée sur les RNN à celle d'approches de l'état de l'art, en oubliant l'aspect séquentiel d'un menu et en le considérant comme un ensemble d'articles. En particulier, nous comparons la performance de notre approche à celle de deux approches : BPR [29] et NeuMF [16].

Nous avons également réalisé une série d'expérimentations sur des modèles de type transformer, qui ont montré qu'ils ne convenaient pas à notre problème.

Nous détaillons ensuite une évaluation expérimentale des aliments prédits dans une séquence, à différents horizons temporels, en utilisant différentes métriques. Pour la classification, nous utilisons la métrique d'*exactitude* (Accuracy) et pour le classement des recommandations, nous utilisons la métrique *Mean Reciprocal Rank* (MRR).

En plus de l'apprentissage séquentiel, nous avons développé différentes variantes d'encodage du contexte et comparé différentes stratégies d'apprentissage avec et sans curriculum learning [7]. La dernière partie de l'article traite de l'utilisation d'un modèle pour prédire des menus complets et d'évaluer la pertinence de ces recommandations.

Cet article présente donc les contributions suivantes :

- une formalisation du problème de recommandation dans le cadre séquentiel de la suggestion de menus ;
- un modèle génératif capable de construire des menus crédibles et personnalisés, en prenant en compte le contexte des utilisateurs ;
- une comparaison précise des performances de notre approche avec celles de bases de référence non séquentielles bien connues et définies ; pour ce faire, nous avons réduit le problème de prédiction de séquence à un simple problème de prédiction d'un ensemble non ordonné ;
- une série d'expérimentations démontrant l'efficacité de l'architecture développée en utilisant des métriques pertinentes contre différentes variantes et ablations.

Les données que nous avons utilisées sont déjà accessibles gratuitement. Tout le code associé à cet article sera mis à disposition en ligne lors de la publication.

## 2 Travaux liés

Les systèmes de recommandation (RS) tirent parti des retours des utilisateurs et des relations entre les items pour suggérer des contenus pertinents. Ces retours peuvent être explicites (par exemple, des évaluations) ou implicites (par exemple, des schémas d'interaction). Les paradigmes les plus courants incluent le filtrage basé sur le contenu, qui repose sur les caractéristiques des items, et le filtrage collaboratif, qui identifie les similarités entre utilisateurs.

Les systèmes de recommandation pour l'alimentation et la nutrition constituent un domaine de recherche actif. Certains modèles utilisent les retours des utilisateurs pour affiner les recommandations, orientant ainsi le système vers des recettes personnalisées [13], tandis que d'autres se concentrent sur la tâche de substitution, consistant à remplacer des ingrédients afin d'améliorer la qualité des menus tout en respectant des contraintes nutritionnelles [1, 3]. Des retours directs ont également été utilisés pour construire des menus personnalisés et équilibrés sur le plan nutritionnel [35].

Les informations contextuelles, telles que des facteurs temporels, géographiques ou émotionnels, peuvent améliorer la personnalisation [31]. Dans les systèmes de recommandation alimentaires, des contextes représentatifs, dérivés d'attributs stables et observables (par exemple, les habitudes de consommation), sont couramment utilisés. Les approches tenant compte du contexte intègrent ces données afin d'optimiser les prédictions.

Les systèmes de recommandation prenant en compte la séquence capturent les dépendances séquentielles entre les items, permettant ainsi de modéliser les habitudes et ten-

dances des utilisateurs [28]. Les réseaux de neurones récurrents (RNN) sont largement utilisés à cet effet, offrant des avantages par rapport aux méthodes traditionnelles en apprenant à partir de sessions complètes d'utilisateurs [17]. Ces systèmes peuvent également capter des intentions à court terme, comme le contexte immédiat des items du menu [27]. Les habitudes de consommation structurées jouent un rôle clé dans les systèmes de recommandation alimentaires. Les contraintes séquentielles — telles que la transition des entrées vers les plats principaux — guident la génération de séquences et améliorent la qualité des recommandations [20]. Par exemple, le choix initial dans un repas (par exemple, céréales et lait) établit des attentes claires pour le reste de la séquence, facilitant ainsi la personnalisation.

Les architectures de type Transformer sont devenues incontournables dans les systèmes de recommandation grâce à leur capacité à modéliser des dépendances à longue portée dans des données séquentielles. Contrairement aux RNN traditionnels, les transformers s'appuient sur des mécanismes d'auto-attention [36], leur permettant de capturer efficacement des motifs complexes. BERT (Bidirectional Encoder Representations from Transformers) [10] a été largement adopté dans les systèmes de recommandation pour sa compréhension contextuelle bidirectionnelle, qui lui permet d'encoder à la fois les interactions passées et futures. Cela le rend efficace pour des tâches telles que la recommandation basée sur les sessions, où l'intention de l'utilisateur peut être déduite des items environnants.

### 3 Les données INCA 2

Les données de consommation alimentaire utilisées dans cet article proviennent de l'enquête INCA2, une enquête nationale française réalisée de 2006 à 2007 sur la consommation alimentaire individuelle. L'enquête se compose de rapports alimentaires documentant la consommation sur 7 jours de 4079 individus, dont 2624 adultes, recueillis sur plusieurs mois afin de tenir compte d'éventuelles variations saisonnières des habitudes alimentaires.

Tous les aliments consommés au cours de la journée sont enregistrés, mais nous avons choisi de nous concentrer sur les trois principaux types de repas : le petit-déjeuner, le déjeuner et le dîner. Les menus sont des séquences d'aliments organisées en 44 groupes. On observe une certaine hétérogénéité dans la manière dont l'eau, le pain et l'huile utilisée lors de la cuisson sont enregistrés dans l'ensemble de données. Pour des raisons de stabilité, nous avons éliminé ces catégories et conservé seulement 34 groupes.

À partir de ces données, nous avons extrait notre jeu de données d'entraînement, qui se compose de 35253 menus, et le jeu de test contenant 6964 menus pour 2502 utilisateurs. Dans l'étude INCA2, plusieurs données contextuelles sont disponibles, en particulier la catégorie d'âge des utilisateurs, qui sera utilisée dans la partie évaluation de cet article.

Dans notre travail, les données de consommation sont interprétées comme les préférences implicites des utilisateurs

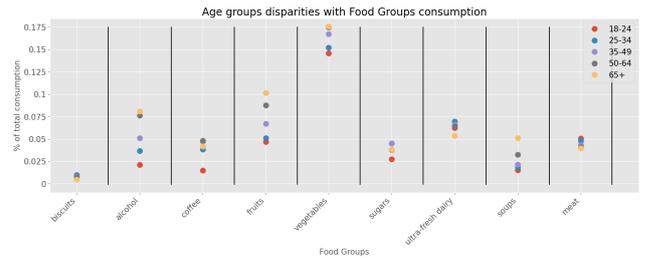


FIGURE 2 – Habitudes de Consommations en fonction de l'Age

teurs [29]. Pour modéliser la consommation des utilisateurs, nous considérons une matrice d'interaction binaire « aliments »/« utilisateurs » où les 0 indiquent la non-consommation de l'aliment par l'utilisateur et les 1 représentent la consommation de l'aliment par celui-ci.

#### Habitudes de Consommations et Contextes.

Nous considérons un menu comme une séquence ordonnée d'items, où chaque item appartient à un groupe alimentaire : une tomate appartient, par exemple, au groupe des légumes. Nous avons dû limiter notre analyse aux groupes alimentaires, car dans ce jeu de données, les aliments individuels sont soit trop rares (par exemple, très peu d'utilisateurs consomment des poivrons) soit trop spécifiques (par exemple, une part importante des utilisateurs consomme de la viande, mais très peu consomment du bœuf bourguignon).

Pour améliorer la pertinence des recommandations que nous générons, nous tirons parti du contexte dans lequel la consommation a eu lieu.

Afin de sélectionner le contexte le plus pertinent, nous avons réalisé une série d'analyses statistiques. Nous avons par la suite examiné les habitudes de consommation au sein des différents groupes alimentaires pour déterminer si un contexte permet de diviser le jeu de données en sous-groupes d'utilisateurs ayant des habitudes de consommation différentes.

Dans la Figure 2, nous pouvons observer que certains utilisateurs présentent des habitudes de consommation différentes pour certains groupes alimentaires tout en étant similaires pour d'autres. Un contexte est sélectionné s'il permet de discriminer les utilisateurs en fonction de leur consommation (c'est-à-dire, s'il existe une quantité suffisante de différences entre leur menus).

De plus, la représentation des différences de consommation entre les groupes d'âge, révèle des disparités significatives. Les personnes du groupe d'âge 18-24 ans consomment moins d'alcool que celles des groupes 50-64 ans et 65 ans et plus. Le café se distingue comme le groupe alimentaire le plus discriminant, les plus jeunes consommant moins que les plus âgés. Ces particularités nous ont conduits à étudier si l'âge pouvait constituer un contexte pertinent.

Par ailleurs, notre analyse a également montré que le type de repas représente un contexte clé à prendre en compte, puisque les repas du petit-déjeuner diffèrent considérablement des autres.

## 4 Modélisation avec des Réseaux de Neurones Récurrents

Il existe de nombreux outils de modélisation séquentielle, allant des chaînes de Markov [30] pour les séquences discrètes aux transformers [21], qui offrent une approche globale pour capturer les dépendances entre des événements éloignés. Nous visons à apprendre des représentations continues pour les utilisateurs et les items alimentaires afin, en définitive, d'étudier à la fois la topologie de la population et la similarité entre les items. Les données avec lesquelles nous travaillons sont de bonne qualité mais limitées, ce qui nous conduit à explorer des *modèles avec peu de paramètres*.

La flexibilité de l'architecture des réseaux de neurones récurrents (RNN) et leur efficacité éprouvée dans les systèmes de recommandation nous incitent à les étudier en détail. De plus, les séquences de consommation sont relativement courtes (4 items en moyenne dans le jeu de données INCA2 après prétraitement) et ne présentent pas toujours des transitions claires, comme entre les types de repas tels que les entrées, les plats principaux et les desserts pour les déjeuners et les dîners.

### 4.1 Données Séquentielles et Contexte

Nous considérons un jeu de données  $X$  étant des menus  $m$ , chaque menu étant une séquence d'éléments alimentaires  $a$  :

$$X = \{m_{u,d,r}\}, \quad m_{u,d,r} = [a_1, \dots, a_t, \dots, a_T] \quad (1)$$

Les menus sont associés à une date  $d$ , un type de repas  $r$  (petit-déjeuner, dîner, *etc.*) et un utilisateur  $u$ . Comme décrit dans la section 3, la date  $d$  est utilisée pour représenter les sept jours de consommation. Nous l'avons exploitée pour diviser les données en ensembles d'entraînement, de validation et de test, et pour différencier les différents menus dans le processus d'apprentissage. L'ensemble d'entraînement est composé de 5 jours de consommation, tandis que les ensembles de validation et de test sont composés d'une seule journée. Ainsi, la date n'est pas mentionnée ni utilisée dans les notations de prédiction, car la paire  $(u, r)$  constitue déjà un identifiant unique dans l'ensemble de test.

### 4.2 Apprentissage de Représentations

Par défaut, l'espace des éléments alimentaires  $\mathcal{A}$  et l'espace des utilisateurs  $\mathcal{U}$  sont discrets. Suivant le paradigme de l'apprentissage de représentations [6], nous projetons ces deux espaces dans un espace vectoriel de dimension  $z$ , avec :

$$a \in \mathcal{A} \mapsto \mathbf{a} \in \mathbb{R}^z, \quad u \in \mathcal{U} \mapsto \mathbf{u} \in \mathbb{R}^z \quad (2)$$

Ces représentations sont initialisées de manière aléatoire et ensuite apprises par l'architecture du réseau de neurones récurrent (RNN) décrite ci-dessous. Cette approche très générale est la plus flexible pour encoder diverses informations, y compris les affinités, dans les profils des utilisateurs et des éléments. Depuis l'émergence des techniques de factorisation de matrice pour la recommandation [24] jusqu'aux

variantes qui ont remporté le défi Netflix [22], de la vague de modélisation des retours implicites [29] aux approches d'apprentissage profond combinant filtrage collaboratif et l'analyse du contenu [40], toutes les avancées récentes sont basées sur l'apprentissage de représentations.

### 4.3 RNN et Séquentialité des Menus

Un RNN met à jour l'état caché au temps  $t$  ( $\mathbf{h}_t \in \mathbb{R}^h$ ) en utilisant la fonction suivante :

$$\mathbf{h}_t = g(W_a \mathbf{a}_t + W_u \mathbf{u}_i + U \mathbf{h}_{t-1}) \quad (3)$$

Ici,  $g$  est une fonction tangente hyperbolique et  $\mathbf{a}_t, \mathbf{u}_i \in \mathbb{R}^z$  sont respectivement les représentations des éléments alimentaires et de l'utilisateur dans l'espace de projection  $\mathbb{R}^z$  (le profil utilisateur est répété à chaque étape pour améliorer les performances).  $\mathbf{h}_{t-1}$  est l'état caché au temps  $t-1$ . Les paramètres de poids sont donnés par les matrices  $W \in \mathbb{R}^{h \times z}$  et  $U \in \mathbb{R}^{h \times h}$ .

Le réseau fournit une prédiction du prochain élément dans une séquence en tant que tâche de classification ; avec un horizon temporel de 1, cela correspond à l'approche teacher-forcing [23]. Au fur et à mesure que l'horizon augmente, la prédiction au temps  $t$  est utilisée comme entrée au temps  $t+1$ , ce que l'on appelle la génération libre. La prédiction  $P(a_{t+1}|a_{1:t})$  est estimée par une fonction softmax :

$$\hat{p} = f(\mathbf{h}_t) = \text{softmax}(V \mathbf{h}_t) \in \mathbb{R}^{|\mathcal{A}|}, \quad V \in \mathbb{R}^{|\mathcal{A}| \times h} \quad (4)$$

$$\hat{a}_{t+1} = \arg \max \hat{p} \quad (5)$$

où la matrice  $V \in \mathbb{R}^{|\mathcal{A}| \times h}$  regroupe les paramètres de prédiction. Le critère d'apprentissage du réseau est généralement l'entropie croisée :  $H(\hat{p}) = -\log \hat{p}(a_{t+1})$ . Pendant l'entraînement, nous utilisons des mini-batches de taille 100 pour économiser du temps de calcul et améliorer le taux d'apprentissage.

Dans la section 5.4, nous définirons les détails de l'implémentation du transformer et montrerons ses performances sur notre jeu de données.

### 4.4 Intégration du Contexte

Pour ajouter de la diversité et de la personnalisation, nous proposons d'ajouter et de combiner des contextes. Dans cette section, nous discuterons de la manière dont nous pouvons combiner les contextes et ce que nous souhaitons accomplir. À chaque paire  $(u_i, m_{u_i,d,r})$ , un ensemble de contextes  $c \in \mathcal{C}$  est associé. Dans ce qui suit, les contextes correspondent à des attributs de l'utilisateur ou du repas : il peut s'agir de la catégorie d'âge ou du type de repas. Les travaux futurs prendront en compte d'autres descripteurs pour encoder la saisonnalité ou les lieux (domicile vs restaurant). Comme pour les éléments et les utilisateurs, le contexte est représenté par des valeurs discrètes et projeté dans un espace latent :  $c \in \mathcal{C} \mapsto \mathbf{c} \in \mathbb{R}^z$ .

Comme nous pouvons le voir dans le Tableau 2, l'utilisation du contexte permet au modèle de mieux comprendre la séquence, mais dans des expériences ultérieures (voir

Tab. 3), nous montrons que le contexte n’aura que peu d’impact après le deuxième ou troisième élément alimentaire, le principal bénéfice étant sur le premier élément. Nous avons exploré plusieurs méthodes pour combiner les différents contextes (concaténation, combinaison linéaire, perceptron multicouche), mais n’avons observé aucune différence significative en termes de performance. Nous présentons donc l’approche la plus simple : la concaténation de chaque contexte en  $c_j$  à chaque étape de temps, ce qui peut être écrit comme

$$\mathbf{h}_t = g(W_a \mathbf{a}_t + W_u \mathbf{u}_i + \sum_j W_{c_j} \mathbf{c}_j + U \mathbf{h}_{t-1}) \quad (6)$$

## 4.5 Métriques Séquentielles

### 4.5.1 Précision Séquentielle

Pour évaluer les prédictions séquentielles du RNN, nous comparons chaque paire d’éléments (prédiction - vérité terrain) dans la séquence. Avec  $m$  un menu de longueur  $n$  et respectivement  $u$  et  $r$  l’utilisateur et le type de repas, la précision séquentielle est définie comme :

$$Acc_{seq}(m_{u,r}) = \frac{1}{n} \sum_{t=1}^n \mathbb{1}_{[a_t=\hat{a}_t]} \quad (7)$$

### 4.5.2 MRR

Compte tenu de la grande diversité de consommation et des relations complexes entre les éléments, se contenter d’une correspondance exacte n’est pas suffisant pour évaluer la qualité d’une prédiction d’un élément. Le Mean Reciprocal Rank (MRR) est une métrique utilisée pour évaluer la qualité des recommandations classées. Elle mesure le rang auquel apparaît le premier élément pertinent dans la liste de recommandations. Nous calculons le MRR pour chaque élément d’un menu en examinant la position de la vérité terrain dans le tenseur de prédiction, puis nous en faisons la moyenne. En notant  $m_{u,r}$  un menu de longueur  $T$  et  $rank_{a_t}$  la position du premier élément pertinent dans la liste de sortie, nous avons :

$$MRR(m_{u,r}) = \frac{1}{|m_{u,r}|} \sum_{t=1}^T \frac{1}{rank_{a_t}} \quad (8)$$

## 4.6 Génération de Séquences

Le processus de génération de séquences utilise la recherche par faisceau (beam search). Au lieu de sélectionner uniquement l’élément le plus probable, plusieurs éléments sont tirés de la distribution de prédiction, en maintenant des séquences parallèles. Cela garantit la diversité et permet d’identifier des séquences optimales qu’une approche avaré pourrait manquer.

La recherche par faisceau permet également une génération contrainte à moindre coût computationnel, en modifiant la distribution pour exclure certains éléments. En prenant en entrée l’utilisateur, le contexte et le début de la séquence, le RNN génère des menus cohérents et personnalisés qui respectent les contraintes nutritionnelles.

Nous avons étudié deux méthodes de prédiction dans notre modèle : (i) *teacher-forcing*, qui utilise la vérité terrain pour prédire à  $t + 1$ , et (ii) *free-generation*, où la prédiction à  $t$  est utilisée comme entrée pour  $t + 1$ .

L’entraînement avec *free-generation* permet d’explorer des séquences plus complexes, améliorant ainsi l’apprentissage, mais commencer l’entraînement de cette manière rend la tâche trop complexe et les gradients instables. Le *teacher-forcing* empêche lui le modèle d’apprendre des dépendances au-delà de  $t + 1$ , comme par exemple éviter la répétition d’éléments dans une séquence. Nous avons implémenté le *scheduled learning* [5], augmentant progressivement l’utilisation de *free-generation* dans les prédictions, mais cela a donné des performances inférieures par rapport à l’utilisation exclusive du *teacher-forcing*.

## 4.7 Baselines et Métriques Non-Séquentielles

Pour ancrer notre travail dans la littérature sur les systèmes de recommandation, nous proposons une série d’expériences non séquentielles qui nous permettent de nous comparer à des baselines solides. Nous avons utilisé BPR [29] et NeuMF [16]. BPR fonctionne avec un système d’optimisation par paires : au lieu de prédire un score pour une paire utilisateur-article, il se concentre sur la comparaison de deux articles pour un utilisateur, fournissant ainsi un classement personnalisé pour chaque paire utilisateur-article. NeuMF utilise la factorisation matricielle généralisée (GMF); au lieu du simple produit scalaire utilisé dans la MF, l’interaction entre les utilisateurs et les articles est modélisée à l’aide d’un produit élément par élément des embeddings des utilisateurs et des articles. Ils combinent leur résultat avec un perceptron multicouche (MLP) qui introduit de la non-linéarité afin de capturer des interactions plus complexes entre utilisateurs et articles.

Les approches BPR et NeuMF ne sont pas capables de prédire des séquences, mais uniquement des listes d’articles : elles ne peuvent pas encoder le fait que la salade est souvent consommée en entrée et donc avant le fromage. De plus, les types de menus ne sont pas pris en compte par ces approches ; nous proposons de construire un modèle pour chaque type de repas (petit-déjeuner, déjeuner, dîner).

Un menu pour un utilisateur  $u$  est désormais défini comme

$$M_u = \{a_1, \dots, a_n\}$$

où  $M$  est un ensemble de  $n$  articles distincts et l’ordre des éléments n’a pas d’importance, c’est-à-dire que

$$\{a_1, a_2\} = \{a_2, a_1\}.$$

Dans ce cadre, un menu se réduit à un ensemble d’aliments et la métrique d’évaluation des performances mesure la précision entre la liste d’aliments proposée par le modèle et la vérité terrain :

$$Acc_{set}(M_u) = \frac{1}{|M_u|} \left| M_u \cap \hat{M}_u \right| \quad (9)$$

Pour BPR et NeuMF, soit  $y \in \mathbf{R}^n$  le tenseur de prédiction de sortie de taille  $n$ , où chaque élément  $y_i$  représente

TABLE 1 – Exactitude sur ensembles pour les modèles NeuMF, BPR et RNN.

Type repas	NeuMF	BPR	RNN
Petit déjeuner	0.616	0.496	<b>0.673</b>
Déjeuner	0.164	0.133	<b>0.197</b>
Dîner	0.168	0.131	<b>0.203</b>
Moyenne tout repas	0.316	0.253	<b>0.357</b>

le score prédit pour le  $i^{\text{ème}}$  article. Nous avons défini une prédiction de menu comme suit :

$$\text{Top}_n(\mathbf{y}) = \{y_1, y_2, \dots, y_n \mid y_1 \geq y_2 \geq \dots \geq y_n\} \quad (10)$$

$$\hat{M}_u = \{a_1, \dots, a_n\} \quad (11)$$

Le RNN est utilisé en free generation et produit une séquence de longueur  $n$  qui est ensuite réduite à un ensemble non ordonné. Par défaut (c’est-à-dire sans introduire de stratégie d’échantillonnage), chaque approche produit une unique prédiction pour chaque type de repas et pour chaque utilisateur. Cette sortie sera comparée à l’ensemble de test, qui est constitué d’une seule journée.

## 5 Expériences sur les données INCA2

Nous proposons d’abord des expériences non-séquentielles (basées sur des ensembles d’articles) afin de comparer notre approche aux références en systèmes de recommandation. Ensuite, nous passons à la génération séquentielle pour analyser l’impact du contexte et la cohérence des prédictions au fur et à mesure de l’avancement du repas.

### 5.1 Comparaison par rapport à l’état de l’art : Baselines non séquentielles

La Table 1 présente les performances sur le jeu de test pour la **exactitude par ensemble** (équation 9). Pour correctement représenter la particularité de notre jeu de données, nous l’avons séparé selon le type de repas (petit déjeuner, déjeuner et dîner), en calculant l’exactitude sur chacun des sous-ensembles séparément (avec 3 modèles différents), *Moyenne tout repas* étant la moyenne des 3 résultats. Comme décrit dans la section 4.7, tous les modèles (**RNN**, **BPR** et **NeuMF**) prédisent un repas complet à partir du profil utilisateur. La disparité de d’exactitude pour le *petit déjeuner* par rapport aux autres types de repas s’explique par le fait qu’il s’agit d’un menu plus facile à prédire, car il comporte moins de diversité d’articles, et il est reconnu comme un menu stable. Nous avons tendance à manger les mêmes choses chaque matin. Les résultats pour le *déjeuner* et le *dîner* montrent que ces menus sont plus diversifiés, et donc plus complexes, ce qui les rend plus difficiles à apprendre avec les quelques données dont nous disposons.

Le modèle **BPR** affiche un score inférieur, ce qui nous indique que la relation complexe entre les articles et les utilisateurs ne peut être pleinement comprise avec une approche linéaire, alors que **NeuMF** et **RNN**, qui utilisent tous deux

TABLE 2 – Performances des modèles pour la prédiction du prochain élément

Modèle	MRR	Acc
(Item)	0.329	0.243
(Item,User)	0.405	0.328
(Item,User,Age)	0.413	0.336
(Item,User,Typm)	<b>0.429</b>	<b>0.355</b>
(Item,User,Typm,Age)	<b>0.429</b>	0.354

un perceptron multicouche, obtiennent de meilleures performances. Compte tenu de la performance globalement supérieure du modèle RNN, nous démontrons que, même sans métrique séquentielle, cette tâche de recommandation est mieux résolue avec une approche séquentielle. En effet, alors que BPR et NeuMF pourraient prédire des articles adaptés à l’utilisateur mais incohérents (par exemple, deux plats principaux sans entrée), le RNN prédit en plus un repas cohérent.

### 5.2 Exploiter le contexte pour améliorer la performance

Comme décrit dans la section 3, l’information contextuelle a un impact significatif sur le contenu des repas des utilisateurs. Nous avons pour objectif de quantifier l’impact du contexte sur des recommandations séquentielles. Nous profitons également de cette expérience pour évaluer l’impact du profil utilisateur sur les performances du système (en traitant l’utilisateur comme un contexte, voir Eq. (6)).

Chaque contexte permet de représenter des sous-groupes d’utilisateurs, la qualité de celui-ci étant déterminée par les différences de consommation entre utilisateurs de sous-groupes différents. Par exemple, si nous utilisons l’âge comme contexte et observons que les personnes de 20 ans consomment moins de café que celles de plus de 60 ans, nous pouvons en déduire que ce contexte pourrait aider le modèle en introduisant un biais en faveur de la consommation de café.

Pour évaluer correctement la qualité d’une recommandation, nous pouvons soit examiner chaque élément individuellement, en mesurant la proximité avec la vérité terrain, soit évaluer la qualité du menu global. Pour mesurer la qualité de chaque prédiction d’élément, nous utiliserons *L’Exactitude Séquentielle (Acc)* et le *Mean Reciprocal Rank (MRR)*.

Les performances de chaque contexte sont présentées dans la Table 2. Nous constatons que la prise en compte de n’importe quel contexte améliore la performance de l’approche, mais seul le type de repas offre une amélioration significative. Nous nous attendions à ce que le contexte lié aux groupes d’âge (*Age*) ne différencie pas les utilisateurs autant que le type de repas (*Typm*).

La performance du contexte âge implique que nous n’avons pas complètement dissocié l’information liée à l’âge de l’embedding utilisateur. De futures expériences tenteront d’extraire autant d’information que possible de cet embed-

TABLE 3 – Disparité d’exactitude des articles en fonction du type de repas. L’exactitude augmente significativement au fur et à mesure du déroulement du déjeuner et du dîner.

Modèle	Petit-déjeuner	Déjeuner	Dîner
Exactitude du premier élément			
(Item,User)	64.8	2.5	3.6
(Item,User,Typm)	75.9	8.0	13.3
Exactitude du deuxième élément			
(Item,User)	63.9	7.5	8.0
(Item,User,Typm)	65.6	9.2	11.3
Exactitude du troisième élément			
(Item,User)	63.5	12.5	16.0
(Item,User,Typm)	63.3	13.6	16.8
Exactitude globale des éléments			
(Item,User)	65.6	20.9	20.7
(Item,User,Typm)	68.4	22.5	24.4

ding utilisateur. Néanmoins, l’ajout d’un contexte supporté par une augmentation des disparités statistiques dans les habitudes de consommation constitue une amélioration.

### 5.3 Analyse Approfondie des Performances des Modèles

Le modèle qui considère le type de repas comme contexte obtient de meilleurs résultats pour le déjeuner et le dîner, mais pas pour le petit-déjeuner. Ce comportement s’explique par le surapprentissage du modèle sur les menus de petit-déjeuner qui, en France, présentent une faible diversité d’aliments et commencent fréquemment par le café.

Nous proposons donc de mesurer la performance en termes d’exactitude du prochain aliment dans le menu. La régularité présente dans la consommation au petit-déjeuner rend l’interprétation des performances difficile pour ce menu. Pour le déjeuner et le dîner, nous constatons qu’il est très difficile de prédire le premier élément (ce qui semble logique compte tenu du nombre important de possibilité des utilisateurs pour cet aliment), mais ensuite les prédictions s’améliorent en tirant parti de l’information qui se dévoile au fil des itérations, ce qui démontre cohérence générale dans les repas. La Table 3 illustre ce phénomène.

Il apparaît également que notre formulation actuelle ne parvient pas à tirer profit de la combinaison des contextes (dernière ligne du tableau). Là encore, il semble s’agir d’un problème de régularisation et/ou de désentrelacement : les paramètres du contexte supplémentaire ne sont pas correctement exploités, l’information étant déjà encodée dans les autres profils.

Néanmoins, nos expériences démontrent l’impact positif de l’utilisation de l’information contextuelle pour améliorer les recommandations dans le cas de données séquentielles. Des travaux antérieurs dans le domaine de la musique et du e commerce [38] abordent l’impact du premier élément sur la génération. Plus tôt le modèle comprend à quel moment du repas il se trouve, meilleure sera la suite de la recomman-

TABLE 4 – Impact du jeu de données sur nos modèles transformer

Jeu de données	MRR	ACC
<b>Modèle (Item)</b>		
(Déjeuner, Dîner)	0.284	0.231
(Petit-déjeuner, Déjeuner, Dîner)	0.356	0.293
<b>Modèle (Item,User)</b>		
(Déjeuner, Dîner)	0.294	0.231
(Petit-déjeuner, Déjeuner, Dîner)	0.366	0.294

dation. Avec le teacher forcing, cet avantage est rapidement compromis, mais avec la génération libre, guider le modèle dès le début de la séquence offre un avantage notable.

### 5.4 RNN vs Transformer

Dans cette section, nous résumons les expériences menées autour du Transformer. Notre implémentation suit de près celle proposée dans "Attention is all you need" [36], en utilisant les modules dédiés de PyTorch. Le modèle est basé sur l’auto-attention bidirectionnelle et est entraîné (et évalué) selon le paradigme du langage masqué (Masked Language Modeling - MLM).

Nous avons exploré plusieurs approches pour introduire l’utilisateur et la personnalisation. Nous avons d’abord ajouté des poids d’utilisateur après chaque bloc de transformation, en concaténant ces poids avec ceux des éléments, comme nous l’avons fait avec le RNN (toutes les entrées sont ainsi doublées en taille). Nous avons également tenté une multiplication terme à terme entre utilisateurs et items afin de modéliser la vision spécifique de l’utilisateur sur l’item, à la manière d’une factorisation matricielle. La méthode la plus efficace consiste à traiter l’utilisateur de la même façon que le token CLS dans BERT [10], en l’ajoutant simplement au début de la séquence.

Après optimisation des hyperparamètres, nous avons utilisé une attention multi-têtes à 8 têtes, avec 4 blocs Transformer et un espace d’embedding pour les utilisateurs et les items de taille 256. Pendant l’entraînement, nous avons utilisé un taux d’apprentissage de 1e-3 et masqué 20% de l’entrée, en entraînant pendant 10 époques (un nombre d’époques supérieur menant à un surapprentissage).

Malgré la prédiction bidirectionnelle sur les aliments masqués, nous constatons dans la Tab. 4 que les performances du modèle Transformer sont inférieures à celles du RNN. Nos expériences montrent également que l’utilisateur n’est pas pris en compte dans la décision : échanger les utilisateurs n’a aucun impact sur la vraisemblance des repas prédits.

Les faibles performances de l’architecture Transformer s’expliquent par le manque de données et la petite taille des repas. L’utilisation d’un mini-transformer ne fonctionne pas, les performances optimales sont obtenues avec une paramétrisation correspondant à des embeddings grands et coûteux, avec un nombre de couches et de têtes qui seraient déraisonnable par rapport à nos données. Nous concluons

ainsi, en accord avec d’autres études [39], que, malgré l’intérêt du mécanisme d’auto-attention pour cette tâche, le Transformer n’est pas adapté à notre cas d’utilisation.

## 6 Analyse des recommandations du modèle

Afin d’analyser notre modèle sous un nouvel angle, nous considérons la log-vraisemblance ; pour un utilisateur  $u_i$  et son menu  $m_{u_i} = \{a_t\}_{t=1, \dots, |m_{u_i}|}$  avec le contexte  $c$ , celle-ci est calculée comme suit :

$$\mathcal{L}(m|u_i, c) = \sum_t \log p(a_t|u_i, c) \quad (12)$$

avec  $p(a_t|u_i, c)$  correspondant aux sorties softmax du RNN, dont les entrées sont les profils  $u_i$  et  $c$ .

Nous utilisons cette métrique pour comparer l’affinité d’un utilisateur pour des menus : soit des menus générés pour lui à l’aide de l’algorithme de beam search, soit des menus consommés par d’autres utilisateurs.

### 6.1 Recommandations personnalisées

Nous avons d’abord exploré différentes manières de combiner et d’utiliser les contextes et évalué notre modèle sur différentes métriques et jeu de données. Cependant aucun de ces résultats ne permet de démontrer que notre modèle est véritablement personnalisé : un modèle peut recommander un article à un utilisateur simplement parce qu’il est populaire et non parce qu’il correspond aux goûts de celui-ci.

Dans une première expérience, nous testons si le modèle recommande des articles de manière générique ou s’il a appris à calibrer la recommandation afin de satisfaire les préférences des utilisateurs. Si le modèle est personnalisé, il faut observer qu’un menu généré pour un utilisateur est plus adapté à un menu aléatoire présent dans le jeu de données, ce dernier n’étant pas directement lié aux goûts de l’utilisateur. Nous sélectionnons un utilisateur au hasard et comparons la log-vraisemblance des menus que nous générons pour lui, via l’algorithme de beam search, à la log-vraisemblance de menus existants provenant d’un autre utilisateur, ces menus étant de taille identique.

Nous calculons le gain moyen de la log-vraisemblance sur 10 menus générés comparativement à 50 menus d’autres utilisateurs (tous les menus ayant une taille de 5 pour permettre l’agrégation et la comparaison des vraisemblances). Nous constatons un gain notable de log-vraisemblance avec notre menu généré, ce qui indique que le modèle est adapté aux goûts des utilisateurs. Il convient également de noter que les menus générés sont assez convaincants pour un humain, même si nous n’avons pas encore mené d’expériences approfondies à ce sujet. Un exemple de menu généré est : [’pâté de campagne’, ’carotte crue’, ’steak haché’, ’pâtes alimentaires cuites’, ’café noir prêt à boire non sucré’].

### 6.2 Application réelle de la personnalisation

La deuxième expérience constitue une application directe de notre modèle à des fins pratiques pour les nutritionnistes. Si, par exemple, une personne doit arrêter de consommer

de la viande, nous pouvons soit recommander un menu vegan préexistant, soit opter pour une génération de menus contraints afin de s’adapter à ce nouveau régime.

Pour cette expérience, nous sélectionnons des utilisateurs qui consomment beaucoup d’un certain groupe d’aliments (soupe, fromage, viande *etc.*). Pour ces utilisateurs, nous générons 10 menus contenant un article issu de ce groupe spécifique à l’aide du beam search. Ensuite, nous générons 10 nouveaux menus en appliquant une contrainte dans le beam search : nous imposons que la génération ne recommande et ne construise que des menus sans l’article sélectionné du groupe alimentaire. Enfin, afin de déterminer si notre recommandation personnalisée correspond mieux aux goûts de l’utilisateur que le menu d’un autre, nous sélectionnons 50 menus aléatoires provenant d’autres utilisateurs, avec et sans l’article sélectionné.

Les résultats sont présentés dans la Fig. 3. Nous souhaitons d’abord comparer le score du menu généré sans l’article sélectionné à celui d’un menu existant dans lequel cette contrainte est respectée. Nous observons un gain significatif de log-vraisemblance pour les deux menus, ce qui implique que notre modèle reste personnalisé même lorsque nous appliquons des contraintes à la recommandation. Cette méthode pourrait être utile pour les nutritionnistes en général et pour l’adaptation de menus à des régimes spécifiques.

## 7 Conclusions

Cet article décrit la conception et l’évaluation d’un système de recommandation séquentielle dans le domaine de la nutrition. Nous mettons en lumière l’importance des données séquentielles dans le domaine de la recommandation alimentaire. Pour répondre aux besoins des experts du secteur, notre modèle est capable de générer des séquences cohérentes d’aliments personnalisées aux goûts des utilisateurs, en tenant compte d’informations contextuelles. Grâce à la mise en place de l’algorithme de beam search contraint, nous pouvons générer des menus adaptés de la taille souhaitée. Nous mesurons les performances de cette architecture et démontrons ses capacités de personnalisation, ce qui nous permet d’envisager des applications concrètes pour modifier efficacement le régime alimentaire d’une per-

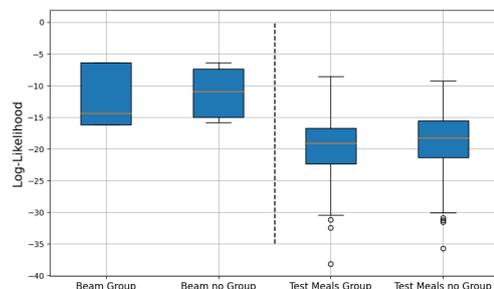


FIGURE 3 – Comparaison entre un menu généré et un menu existant, avec ou sans un groupe d’aliments spécifique non consommé par l’utilisateur

sonne, tout en continuant à recommander des menus adaptés à ces goûts.

Les architectures basées sur le Transformer semblent inadaptées en raison de la faible quantité de données dont nous disposons. Compte tenu du coût computationnel de ces modèles, nous privilégions également des modèles légers et économes, tels que le RNN. Nous devons admettre que le problème du désentanglement du contexte d'âge et de l'intégration de l'utilisateur n'a pas été complètement résolu.

Une autre amélioration sur laquelle nous travaillons est l'intégration d'un graphe de connaissances nutritionnelles dans le processus d'apprentissage, afin de mesurer plus précisément la qualité nutritionnelle de la prédiction d'un menu complet. Ce graphe de connaissances pourra également être utilisé pour créer des filtres plus complexes lors du beam search.

## Références

- [1] Palakorn Achananuparp and Ingmar Weber. Extracting food substitutes from food diary via distributional similarity, 2016.
- [2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*. Springer, 2010.
- [3] Sema Akkoyunlu, Cristina E. Manfredotti, Antoine Cornuéjols, Nicolas Darcel, and Fabien Delaere. Investigating substitutability of food items in consumption data. CEUR-WS.org, 2017.
- [4] ANSES. Données de consommations et habitudes alimentaires de l'Étude inca 2, 2014.
- [5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015.
- [6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning : A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. ACM, 2009.
- [8] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. Recsys challenge 2018 : Automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 527–528, 2018.
- [9] Imam Cholissodin and Ratih Kartika Dewi. Optimization of healthy diet menu variation using pso-sa. *Journal of Information Technology and Computer Science*, 2017.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding, 2019.
- [11] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. A hybrid collaborative filtering model with deep structure for recommender systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.
- [12] David Elsweiler and Morgan Harvey. Towards automatic meal plan recommendations for balanced nutrition. In *Proceedings of ACM RecSys*. ACM, 2015.
- [13] Mouzhi Ge, Francesco Ricci, and David Massimo. Health-aware food recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015*. ACM, 2015.
- [14] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-commerce in your inbox : Product recommendations at scale. In *ACM SIGKDD*, 2015.
- [15] Elie Guàrdia-Sebaoun, Vincent Guigues, and Patrick Gallinari. Latent trajectory modeling : A light and efficient way to introduce time in recommender systems. In *ACM RecSys*, 2015.
- [16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering, 2017.
- [17] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top- $k$  gains for session-based recommendations. pages 843–852. ACM, 2018.
- [18] Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv :1704.07138*, 2017.
- [19] Barbara J. Ivens and Marianne Smith Edge. Translating the dietary guidelines to promote behavior change : Perspectives from the food and nutrition science solutions joint task force. *J Acad Nutr Diet*, 2016.
- [20] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. Beyond "hitting the hits" : Generating coherent music playlist continuations with the right tracks. ACM, 2015.
- [21] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018.
- [22] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix Prize Documentation*, 2009.
- [23] Alex M Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing : A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, 2016.
- [24] Daniel Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13, 2000.
- [25] Wee Sun Lee. Collaborative learning for recommender systems. In *ICML*, 2001.

- [26] World Health Organization. World health organization. diet, nutrition and the prevention of chronic diseases : Report of a joint who/fao expert consultation. Technical report, World Health Organization, 2003.
- [27] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. Recommendation systems with complex constraints : A courserank perspective. *ACM Trans. Inf. Syst.*, 2011.
- [28] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Computing Surveys*, 2018.
- [29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr : Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv :1205.2618*, 2012.
- [30] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.
- [31] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems : Introduction and challenges. *Recommender Systems Handbook*, 2015.
- [32] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, 1999.
- [33] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments : Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. JMLR.org, 2016.
- [34] Harald Steck. Evaluation of recommendations : Rating-prediction and ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*, pages 213–220. Association for Computing Machinery, 2013.
- [35] Raciél Yera Toledo, Ahmad A. Alzahrani, and Luis Martínez. A food recommender system considering nutritional information and user preferences. *IEEE Access*, 2019.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [37] Densie Webb and Carol Byrd-Bredbenner. Overcoming consumer inertia to dietary guidance. *Advances in Nutrition*, 2015.
- [38] Jie Xu, Tianwei Xing, and Mihaela van der Schaar. Personalized course sequence recommendations. *IEEE Transactions on Signal Processing*, 2016.
- [39] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [40] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system : A survey and new perspectives. *ACM Computing Surveys*, 2019.