



**HAL**  
open science

## Un système de recommandation social pour des recommandations personnalisées de recettes

Florian Pecune, Lucile Callebert, Stacy Marsella

### ► To cite this version:

Florian Pecune, Lucile Callebert, Stacy Marsella. Un système de recommandation social pour des recommandations personnalisées de recettes. Workshop sur les Affects, Compagnons artificiels et Interactions, CNRS, Université Toulouse Jean Jaurès, Université de Bordeaux, Jun 2020, Saint Pierre d'Oléron, France. hal-02933488

**HAL Id: hal-02933488**

**<https://inria.hal.science/hal-02933488>**

Submitted on 8 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Un système de recommandation social pour des recommandations personnalisées de recettes

Florian Pecune  
University of Glasgow, UK  
florian.pecune@glasgow.ac.uk

Lucile Callebert  
University of Glasgow, UK  
lucile.callebert@glasgow.ac.uk

Stacy Marsella  
University of Glasgow, UK  
stacy.marsella@glasgow.ac.uk

## RÉSUMÉ

Une solution potentielle pour aider les gens à changer leur comportement alimentaire consiste à développer des systèmes conversationnels capables de recommander des recettes saines. Au-delà de la qualité intrinsèque de la recommandation elle-même, divers facteurs peuvent influencer la perception qu'ont les utilisateurs d'une recommandation. Deux de ces facteurs sont le style conversationnel du système et la modalité d'interaction avec ses utilisateurs. Dans cet article, nous présentons Cora, un système conversationnel qui recommande des recettes alignées sur les habitudes alimentaires et les préférences actuelles de ses utilisateurs. Les utilisateurs peuvent interagir avec Cora en cliquant sur des boutons ou en écrivant du texte en langage naturel. De son côté, Cora peut engager les utilisateurs via un dialogue social ou aller droit au but. Nous avons mené une expérience pour évaluer l'impact du style conversationnel de Cora et du mode d'interaction avec les utilisateurs sur la qualité perçue de l'interaction et du système, et sur l'intention de l'utilisateur de cuisiner les recettes recommandées. Nos résultats montrent qu'un système de recommandation conversationnel qui engage ses utilisateurs à travers un dialogue social améliore la qualité perçue de l'interaction ainsi que celle du système.

## KEYWORDS

système de recommandation; agent conversationnel social

## 1 INTRODUCTION

Manger sain implique des processus de prise de décision complexes [6] qui nécessitent d'être conscient des différentes options et de faire le bon choix parmi différentes recettes [21]. Une solution pour surmonter ce problème et aider les gens à faire des choix alimentaires plus sains consiste à développer des systèmes de recommandation de recettes [26]. Au-delà de la qualité de la recommandation elle-même, de nombreux autres facteurs peuvent influencer l'expérience globale des utilisateurs lorsqu'ils interagissent avec un système de recommandation [13]. L'une des directions envisagées pour améliorer l'expérience utilisateur est d'intégrer le processus de recommandation dans une approche conversationnelle [20]. Un agent conversationnel peut en effet aider son utilisateur à accomplir sa tâche (e.g. trouver une recette) mais aussi établir du *rapport* : une bonne relation interpersonnelle ou "chimie" établie lors d'une interaction [24, 25]. Des études d'interactions humain-humain ont montré que le rapport a un impact positif sur la performance liée à la tâche dans des domaines divers (e.g. tutorat par des pairs [22], négociation [7]). Il paraît donc judicieux de doter les systèmes de

recommandation de capacités conversationnelles leur permettant d'établir du rapport avec leur utilisateur.

Nous présentons dans ce papier un système conversationnel capable de recommander des recettes à ses utilisateurs tout en construisant du rapport avec eux. Plus spécifiquement, nous nous intéressons ici à la manière dont les stratégies conversationnelles employées par le système ainsi que les modes d'interaction offerts à l'utilisateur influencent la perception du système par ces derniers et leur intention de cuisiner la recette recommandée. Dans les sections suivantes, nous présentons d'abord un état de l'art du domaine, puis nous détaillons l'architecture de notre système et le processus de recommandation. Nous décrivons ensuite l'expérience impliquant des utilisateurs humains que nous avons menée pour évaluer l'impact des stratégies conversationnelles et des modes d'interaction sur la force de persuasion de notre système. Enfin, nous exposons et analysons les résultats de cette expérience avant de conclure.

## 2 ÉTAT DE L'ART

*Systèmes de recommandation de recettes.* De nombreux efforts ont été faits durant ces dernières années pour optimiser les algorithmes de recommandation de recettes. L'une des approches les plus courantes dans ce domaine est de recommander une recette en fonction des ingrédients qu'elle contient. Dans [8], par exemple, les auteurs ont développé un système qui infère les ingrédients préférés d'un utilisateur en fonction des recettes qu'il aime. Le système recommande alors de nouvelles recettes contenant les ingrédients inférés précédemment. Une autre approche couramment utilisée s'appuie sur les qualités nutritionnelles des recettes. YumMe, le système de recommandation développé dans [30], extrait automatiquement les informations nutritionnelles d'une recette à partir de l'image de cette dernière. Le système se sert alors de ces informations pour définir un profil de chaque utilisateur et recommander des recettes correspondant à leur profil. Dans [10], les auteurs analysent les avis que les utilisateurs donnent après chaque recommandation pour mieux comprendre leur comportement et leur raisonnement. Selon les auteurs, les participants peuvent être regroupés en deux catégories : ceux désirant des recettes saines, et ceux pour qui cela n'a pas d'importance. Les analyses révèlent que les deux principales caractéristiques permettant de prédire l'avis relatif à une recette d'un utilisateur de la première catégorie sont la teneur calorifique et le taux de lipides de cette recette.

Ces travaux se consacrent uniquement à l'amélioration des algorithmes de recommandation dans le but de délivrer des recommandations qui plaisent à leurs utilisateurs. Aucun de ces travaux ne s'intéresse à la manière avec laquelle les utilisateurs interagissent avec les systèmes, et vice-versa. Or, cela peut avoir un impact sur l'expérience de ces mêmes utilisateurs [14].

*Systèmes de recommandation conversationnels.* Comme indiqué par [20], une manière d'améliorer l'expérience utilisateur est de doter les systèmes de recommandation de capacités conversationnelles. Une formalisation est proposée par [29], où les auteurs définissent deux phases lors d'une interaction avec un système de recommandation. La phase d'interview consiste en une séquence de questions dans le but de collecter les préférences de l'utilisateur. Le but de la phase de recommandation est de délivrer une ou plusieurs recommandations correspondant aux préférences précédemment collectées. Plus récemment, plusieurs approches telles que [31] ont cherché à optimiser ces deux phases en apprenant au système quelles questions poser avant une recommandation pour maximiser les chances d'obtenir un avis positif de la part de l'utilisateur. Au lieu de s'intéresser à la manière dont le système interagit avec ses utilisateurs, [17] s'intéresse à la manière dont les utilisateurs interagissent avec le système. L'expérience menée par les auteurs montre que l'utilisation de boutons pour désambigüiser une requête écrite en langage naturel permet au système de délivrer de meilleures recommandations.

Ces systèmes ont pour unique objectif d'aider les utilisateurs à accomplir leur tâche lors de l'interaction. Cependant, les humains cherchent également à nouer des relations avec leurs interlocuteurs et à construire du rapport avec ces derniers [25].

*Systèmes conversationnels construisant du rapport.* Avec l'agent immobilier virtuel Rea [2], des chercheurs ont étudié l'influence du *small-talk*<sup>1</sup> sur le prix qu'un utilisateur est prêt à investir dans une maison. Le robot guide de musée Tinker a prouvé que les visiteurs retiennent mieux les informations qui leur sont données si des stratégies conversationnelles sociales sont employées pendant la conversation [1]. Les auteurs de [15] ont montré que dans le contexte d'un système de recommandation conversationnel, communiquer des informations personnelles et partageant les préférences de l'utilisateur a un impact positif sur la satisfaction de l'utilisateur avec le système et sur son intention d'utiliser le système dans le futur. Récemment, les utilisateurs de l'agent de recommandation de films décrit dans [18] ont trouvé les recommandations meilleures lorsque l'agent utilisait ses opinions et expériences "personnelles" pour justifier ses recommandations.

Bien que les stratégies conversationnelles pour construire du rapport soient au cœur de ces travaux, peu d'entre eux ont évalué l'impact de ces stratégies sur la perception de la recommandation par l'utilisateur. L'impact sur l'utilisateur du mode d'interaction avec l'agent n'est pas non plus évalué. Nous présentons donc dans ce papier un système de recommandation conversationnel capable de construire du rapport avec ses utilisateurs en employant différentes stratégies conversationnelles, et offrant différents modes d'interactions à ses utilisateurs. Plus précisément, nos questions de recherche sont les suivantes :

**QR1** : Comment les modes d'interaction entre un agent et son utilisateur influencent-ils la qualité perçue de l'interaction et du système, et l'intention de cuisiner la recette recommandée ?

**QR2** : Comment les stratégies conversationnelles employées par l'agent influencent-elles la qualité perçue de l'interaction et du système, et l'intention de cuisiner la recette recommandée ?

## 3 DESIGN DE L'AGENT CONVERSATIONNEL

Pour répondre à nos questions de recherche nous avons créé et déployé Cora, un agent conversationnel capable de recommander des recettes tout en établissant du rapport avec ses utilisateurs. Nous décrivons ici le design et l'architecture de notre système ainsi que le processus de recommandation.

### 3.1 Design du système

L'interaction entre Cora et l'utilisateur suit le classique scénario interview - recommandation [29]. Pour identifier les questions que Cora devrait poser lors de la phase d'interview, nous avons mené une étude pilote avec environ 100 participants. Après s'être présentée, Cora demande à l'utilisateur : s'il a faim; s'il souhaite manger sain; s'il suit un régime particulier (e.g. vegan) ou s'il a des intolérances (e.g. au gluten); combien de temps il souhaite consacrer à la préparation du repas; s'il y a des ingrédients spécifiques qu'il souhaite utiliser. Cora entre ensuite dans la phase de recommandation. Pour chaque recette recommandée, l'utilisateur peut accepter ou refuser la recommandation et Cora met à jour le profil de l'utilisateur en fonction de la réponse. Cora demande ensuite à l'utilisateur s'il veut une autre recommandation. En cas de réponse positive, Cora recommande une autre recette, alors qu'en cas de réponse négative, Cora termine poliment la conversation.

Pour que Cora construise du rapport avec son utilisateur, nous avons implémenté les stratégies conversationnelles sociales suivantes extraites de [32] :

*Small Talk.* Dans la vie courante, le *small-talk* est utilisé pour briser la glace avec quelqu'un [23]. Dans notre implémentation, Cora pose à l'utilisateur quatre questions en début d'interaction, lui demandant : comment il s'appelle; comment il va; ce qu'il mange habituellement au dîner; pourquoi il mange cela.

*Communication d'informations personnelles.* Communiquer des informations personnelles à son interlocuteur permet d'établir une relation de confiance [16]. Cora révèle donc à son utilisateur des informations sur elle-même lors du *small-talk*. Par exemple, elle explique qu'elle essaye de manger sain et qu'elle adore cuisiner.

*Feedbacks et préférences partagées.* Donner un feedback à l'utilisateur (e.g. *okay*, *d'accord*, *c'est noté*, etc.) permet à Cora de montrer qu'elle écoute activement [27]. De plus pour construire du rapport, Cora approuve les choix de l'utilisateur et partage les préférences de ce dernier [11] : si l'utilisateur dit avoir faim, elle répond qu'elle a faim aussi.

*Opinions personnelles.* D'après [18], il est possible pour un système de construire du rapport avec son utilisateur en justifiant ses choix par des opinions personnelles. Ici, Cora explique son choix de recette en disant par exemple que c'est sa favorite ou qu'elle la cuisine très souvent.

### 3.2 Architecture

*3.2.1 Clients.* Une simple page web permet aux clients de se connecter à notre serveur et de chatter avec Cora. Les messages des utilisateurs et de Cora sont affichés dans une fenêtre de défilement vertical. Lorsque Cora recommande une recette, un poster contenant une image du plat, les ingrédients et les instructions est affiché.

1. Conversation courte et polie qui n'aborde aucun sujet important.

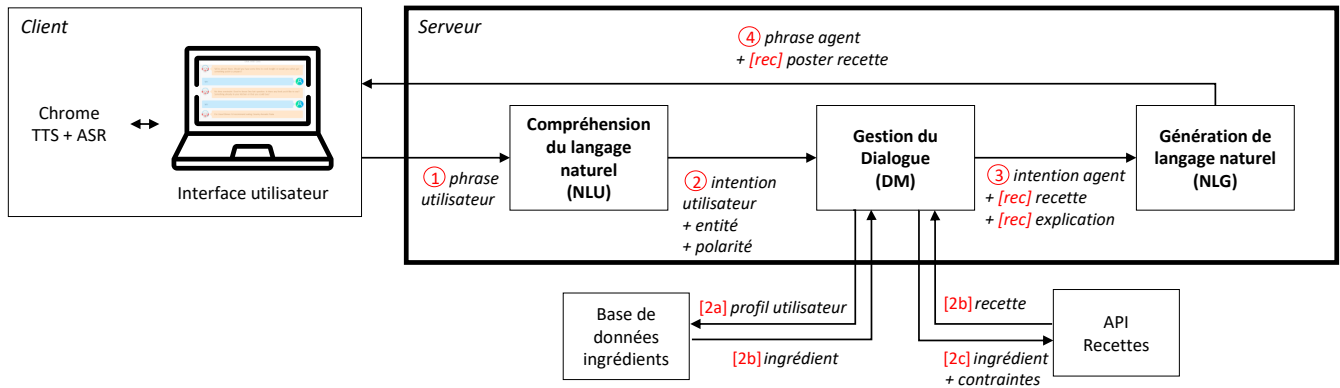


FIGURE 1: Architecture de Cora. Les éléments relatifs au processus de recommandation sont indiqués entre crochets.

Nous avons créé deux interfaces différentes pour l'utilisateur (**User-Mode**). En *chat-mode*, l'utilisateur est libre d'entrer directement du texte en langage naturel dans un champ texte. En *buttons-mode*, l'utilisateur sélectionne un message pré-écrit en cliquant sur des boutons ou en sélectionnant une option dans une liste déroulante (voir figure 2). Par exemple, à la question "Voulez-vous une autre recette?" l'utilisateur peut répondre "Oui" ou "Non merci".

3.2.2 *Serveur*. Le serveur a été développé en Python et gère les connexions en créant un agent Cora dédié pour chaque nouveau client. Chaque agent Cora est composé de trois modules :

1) *Module de compréhension du langage naturel (NLU)*. Le NLU extrait des intentions communicatives, des entités, et la polarité associée aux entités à partir du message de l'utilisateur. Par exemple, avec en entrée le message "I don't like mushrooms", le NLU donnera en sortie {intent: "inform", entity\_type: "food", entity: "mushroom", valence: "-"} . Pour ce faire, nous utilisons les bibliothèques Python Spacy et nltk pour lemmatiser, parser les dépendances, et extraire les catégories grammaticales de chaque token. À partir de ces éléments, un ensemble de règles nous permet d'extraire les intentions communicatives, les entités et la polarité.

2) *Module de gestion du dialogue (DM)*. Prenant en entrée les informations venant du NLU, le DM enregistre les données utilisateur dans un profil utilisateur de manière à y avoir accès lors de la phase de recommandation. Le DM doit ensuite décider du contenu de la réponse à envoyer à l'utilisateur. Nous utilisons pour cela une machine à états finis dont les états correspondent à l'état courant du dialogue et les transitions aux messages de l'utilisateur.

3) *Module de génération de langage naturel (NLG)*. Le NLG utilise toutes les informations reçues des deux précédents modules pour générer une réponse en langage naturel via une table de correspondance. Nous avons défini deux modes d'interaction pour Cora (**Cora-Mode**) : dans le mode tâche *task-conv*, Cora est exclusivement concentrée sur sa tâche de recommandation, alors que dans le mode social *social-conv*, Cora utilise toutes les stratégies définies en 3.1 pour construire du rapport avec son utilisateur.

### 3.3 Processus de recommandation

Les valeurs perçues de *healthiness* et de *fillingness* d'un aliment (cet aliment est-il en général perçu comme sain? comme nourrissant?) influencent fortement la fréquence à laquelle cet aliment est consommé [4] et également le taux d'acceptation de recommandation de recettes contenant cet aliment [10]. Notre processus de recommandation repose donc sur ces deux dimensions. Dans un premier temps, Cora établit un profil utilisateur en évaluant à la fois les habitudes alimentaires et les besoins courants de l'utilisateur en termes de *healthiness* et *fillingness*. Cora sélectionne ensuite un ingrédient puis une recette correspondant à ce profil. Nous avons collecté une base de données contenant 40 ingrédients fréquemment consommés pour le dîner. À chaque ingrédient sont associées une valeur de *healthiness* et une valeur de *fillingness* comprises dans l'intervalle  $[-1; 1]$ . Ces valeurs ont été obtenues en calculant la moyenne des scores assignés par plusieurs centaines de participants.

3.3.1 *Habitudes alimentaires*. Les habitudes alimentaires de chaque participant sont évaluées de la manière suivante : l'utilisateur rapporte la fréquence  $freq_j$  à laquelle il consomme sept ingrédients  $j$  différents sur une échelle de Likert à sept points (de *never* à *every day*) dont les scores sont normalisés dans l'intervalle  $[-1, 1]$ . Ces ingrédients sont extraits de notre base de données : nous avons sélectionné les deux ingrédients ayant les valeurs les plus élevées de *healthiness* et les deux ayant les valeurs les plus faibles de *healthiness*. Nous avons fait de même pour *fillingness*, obtenant ainsi deux ensembles de quatre éléments. Ces ensembles ayant un élément en commun, l'union des deux contient sept éléments.

Pour calculer le score d'habitude alimentaire lié à la *healthiness*  $p(t)_h \in [-1, 1]$ , nous calculons d'abord, pour chaque ingrédient  $j$  un score  $s_{h,j} = freq_j$  si  $j$  est un ingrédient sain (i.e. dont la valeur de *healthiness* est positive) ou  $s_{h,j} = -freq_j$  si  $j$  n'est pas un aliment sain. Nous calculons ensuite la moyenne de ces scores :  $p(t)_h = (\sum_j s_{h,j})/7$ . Nous procédons de manière similaire pour obtenir le score d'habitude de *fillingness*  $p(t)_f$ .

Ces données sont stockées dans le profil utilisateur du DM et sont disponibles pendant toute la durée de l'interaction.

3.3.2 *Besoins courants.* Cora évalue les besoins courants en termes de healthiness et fillingness de son utilisateur en posant les questions "How healthy do you want your meal to be?" et "How hungry are you?" respectivement. Le NLU extrait de la réponse de l'utilisateur des valeurs  $p(s)_h$  (resp.  $p(s)_f$ ) dans l'ensemble  $\{-1; -0.75; -0.5; -0.25; 0; 0.25; 0.5; 0.75; 1\}$ . Par exemple, "no preference" correspond à une valeur de 0 alors que "slightly" correspond à une valeur de 0.25.

Les préférences finales de l'utilisateur sont calculées comme la moyennes des valeurs d'habitudes et de besoins courants, résultant en deux valeurs  $p_h$  et  $p_f$  comprises dans l'intervalle  $[-1; 1]$ .

En plus de ces valeurs, notre étude pilote a révélé que trois autres éléments sont critiques pour que l'utilisateur soit satisfait de la recommandation : il faut tenir compte de ses éventuelles intolérances ou régimes alimentaires (e.g. vegan), du temps qu'il souhaite consacrer à la préparation du repas et des ingrédients dont il dispose. Ces informations sont collectées par Cora lors de la phase d'interview.

3.3.3 *Recommandation.* Une fois la phase d'interview terminée, Cora recommande une ou plusieurs recette à son utilisateur suivant le protocole décrit ci-dessous :

1) *Sélection d'un ingrédient.* Dans le cas où l'utilisateur ne fournit pas d'ingrédient, Cora sélectionne celui qui correspondrait le mieux à ses préférences en utilisant notre base de données d'ingrédients. Pour cela, les préférences de l'utilisateur sont représentées par le vecteur  $p = \langle p_h, p_f \rangle$ . Chaque ingrédient de la base de données est représenté dans le même espace vectoriel, et les ingrédients sont triés dans la liste *ingredients\_list* en fonction de leur distance à  $p$ , du plus proche (i.e. du plus similaire aux préférences de l'utilisateur) au plus éloigné. Les ingrédients que l'utilisateur n'aime pas ou ne peut consommer (e.g. du poulet pour un utilisateur vegan) sont exclus de la liste.

2) *Sélection d'une recette.* Nous utilisons l'API Spoonacular<sup>2</sup> pour trouver des recettes pour nos utilisateurs. Cette API nous permet de spécifier des ingrédients à inclure ou à exclure de la recette, un temps de préparation et un régime alimentaire. Chaque recette est décrite par un titre, une liste d'ingrédients, des instructions et des valeurs nutritionnelles, notamment un score de healthiness et une quantité de calories, que nous transformons en score de fillingness. Cora envoie une requête demandant deux résultats contenant le(s) ingrédient(s) choisis pour/par l'utilisateur et stocke ces résultats dans une *recipes\_list*. De la même manière que pour les ingrédients, les éléments de *recipes\_list* sont représentés par leur vecteur  $p(r) = \langle p(r)_h, p(r)_f \rangle$  dans l'espace vectoriel healthiness-fillingness et triés selon leur distance à  $p$ .

3) *Utilisation des retours-utilisateurs.* Si Cora doit faire plus d'une recommandation, elle adapte le choix de recette aux éventuels commentaires de l'utilisateur. Si ce dernier a précisé ne pas aimer un ingrédient (resp. recette), les recommandations suivantes ne contiendront pas cet ingrédient.

## 4 EXPÉRIENCE

Pour répondre à nos questions de recherche **QR1** et **QR2**, nous avons mis en place une expérience visant à mesurer l'influence du style de conversation et du mode d'interaction sur la qualité perçue de l'interaction et du système, ainsi que sur l'intention de cuisiner les recettes recommandées.

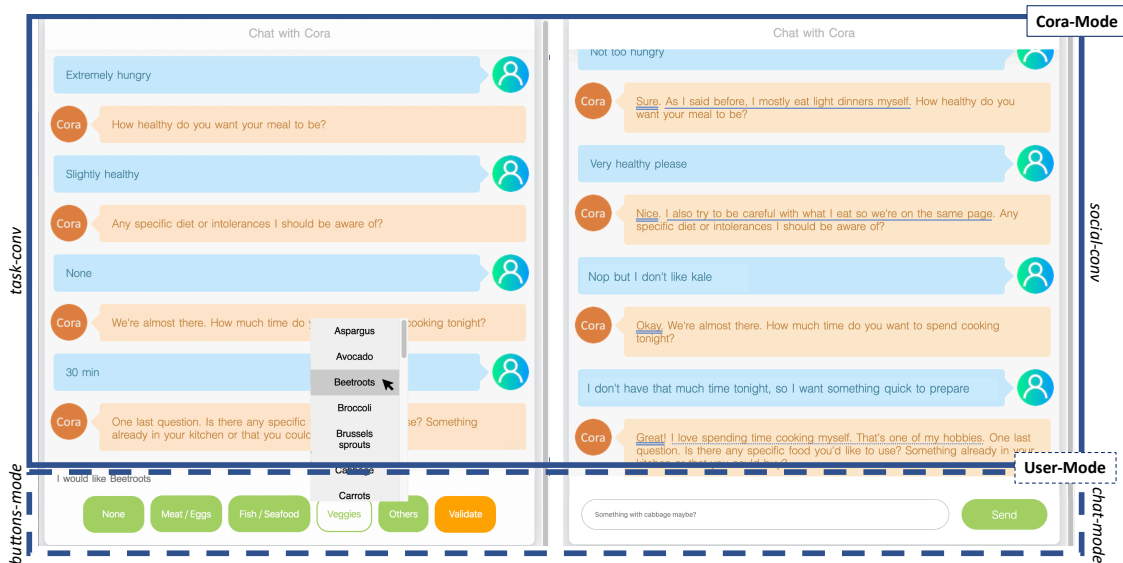
### 4.1 Stimuli

Pour cette expérience, nous avons identifié deux variables indépendantes qui correspondent aux deux modes de fonctionnement du NLG et aux deux types d'interfaces décrits dans la section 3.2. La première variable représente le style de conversation de Cora (**Cora-Mode**). Cette variable a deux niveaux : dans la version sociale (*social-conv*) Cora fait vivre la conversation en utilisant des stratégies conversationnelles permettant de construire du rapport avec l'utilisateur, tandis qu'au cours de la conversation tâche (*task-conv*) Cora va à l'essentiel simplement en posant des questions et recommandant des recettes. La seconde variable (**User-Mode**) représente la manière dont les utilisateurs peuvent interagir avec Cora, et a aussi deux niveaux : dans le mode bouton (*button-mode*), les utilisateurs interagissent avec Cora via des boutons et des menus déroulants, alors que dans le mode chat (*chat-mode*) les utilisateurs interagissent en écrivant en langage naturel. Notre expérience suit un design 2x2 avec **Cora-Mode** et **User-Mode** en tant que variables inter-sujets. Deux exemples d'interactions dépeignant les différences entre les conditions sont représentés par la figure 2.

Dans chacune des quatre conditions, les participants suivaient la même procédure. Les participants ayant lu et accepté notre formulaire de consentement devaient ensuite lire une description de la tâche situant le contexte de l'interaction. Avant d'interagir avec Cora, les participants devaient remplir un questionnaire à propos de leurs habitudes culinaires tel que décrit en section 3.3.1. Chaque participant se voyait alors assigné à une condition expérimentale et interagissait avec Cora selon le scénario décrit en section 3.1. À la fin de l'interaction, les participants devaient remplir trois questionnaires portant sur la qualité de l'interaction, du système, et sur leur intention de cuisiner les recettes recommandées. En plus de ces trois questionnaires, nous posons deux questions ouvertes portant sur Cora et l'expérience en général. Pour finir, les participants devaient répondre à un questionnaire démographique. L'expérience a été validée par le comité d'éthique de l'université de Glasgow.

Pour cette expérience, nous avons mesuré les trois notions suivantes : (1) nous nous appuyons sur la notion de rapport [24] comme un proxy pour mesurer la qualité de l'interaction. Les huit différents éléments que nous avons utilisé pour évaluer le rapport sont inspirés de [9, 33] et listés dans la table 1. (2) La qualité du système de recommandation est mesurée via un questionnaire contenant sept éléments adapté de [19]. Nous avons également ajouté un huitième élément pour évaluer si les participants percevaient les recettes recommandées comme étant saines ou non. Les huit items sont listés dans la table 2. Pour finir, nous mesurons l'intention de cuisiner les recettes recommandées via un questionnaire adapté de [5] dont les cinq éléments sont listés dans la table 3. Les participants répondaient aux questions via une échelle de Likert à 7 points allant de 0 (pas du tout d'accord) à 7 (tout à fait d'accord).

2. <https://spoonacular.com/food-api>



**FIGURE 2: Extraits de deux interactions entre Cora et des participants montrant les deux styles de conversation (*task-conv* vs *social-conv*) et les deux modes d'interaction (*buttons-mode* vs *chat-mode*) utilisés dans notre expérience. Les stratégies conversationnelles sociales utilisées par Cora dans le mode *social-conv* sont soulignées dans le dialogue.**

## 4.2 Hypothèses

Les travaux décrits par [15, 18] nous ont appris qu'un système utilisant des stratégies conversationnelles sociales avait une influence positive sur la satisfaction des utilisateurs et sur leur intention d'utiliser le système. Nous nous attendons donc à trouver un impact positif similaire du style de conversation de Cora sur la qualité du système, de l'interaction et sur l'intention de cuisiner. Nos hypothèses sont les suivantes :

**H1-a :** Le style de conversation du système (**Cora-Mode**) aura un effet principal sur la qualité perçue de l'interaction. Plus précisément, les interactions avec un système capable d'engager les participants en utilisant des stratégies conversationnelles sociales (*social-conv*) seront perçues comme étant meilleures que les interactions avec un système se concentrant uniquement sur l'aspect tâche de l'interaction (*task-conv*).

**H1-b :** Le style de conversation du système (**Cora-Mode**) aura un effet principal sur la qualité perçue du système. Plus précisément, un système capable d'engager les participants en utilisant des stratégies conversationnelles sociales (*social-conv*) sera perçu comme plus performant qu'un système se concentrant uniquement sur l'aspect tâche de l'interaction (*task-conv*).

**H1-c :** Le style de conversation du système (**Cora-Mode**) aura un effet principal sur l'intention de cuisiner des participants. Plus précisément, les participants interagissant avec un système capable de les engager en utilisant des stratégies conversationnelles sociales (*social-conv*) seront plus favorables à l'idée de cuisiner les recettes

recommandées que les participants interagissant avec un système se concentrant sur l'aspect tâche de l'interaction (*task-conv*).

Construire une relation ou du rapport avec quelqu'un est un processus dyadique qui demande aux deux interlocuteurs de s'impliquer de manière équivalente dans l'interaction. Nous nous attendons donc à ce que la possibilité de chatter avec le système en langage naturel améliore l'expérience de l'utilisateur. Nos hypothèses sont les suivantes :

**H2-a :** Le mode d'interaction (**User-Mode**) aura un effet principal sur la qualité perçue de l'interaction. Plus précisément, les interactions des participants ayant la possibilité de chatter avec le système en langage naturel (*chat-mode*) seront perçues comme meilleures que celles des participants interagissant à l'aide de boutons et de menus déroulants (*buttons-mode*).

**H2-b :** Le mode d'interaction (**User-Mode**) aura un effet principal sur la qualité perçue du système. Plus précisément, les participants ayant la possibilité de chatter en langage naturel (*chat-mode*) percevront le système comme plus performant comparativement aux participants interagissant à l'aide de boutons et de menus déroulants (*buttons-mode*).

**H2-c :** Le mode d'interaction (**User-Mode**) aura un effet principal sur l'intention de cuisiner des participants. Plus précisément, les participants ayant la possibilité de chatter avec le système en langage naturel (*chat-mode*) seront plus favorables à l'idée de cuisiner les recettes recommandées que les participants interagissant à l'aide de boutons et de menus déroulants (*buttons-mode*).

Dimensions	Questions	Cora-Mode		User-Mode	
		task-conv	social-conv	button-mode	chat-mode
Coordination	I felt I was in sync with Cora.	4.58(±1.72)	4.85(±1.07)	4.77(±1.22)	4.66(±1.62)
	I was able to say everything I wanted to say during the interaction.	4.81(±1.57)	5.00(±1.33)	4.77(±1.52)	4.94(±1.39)
Attention Mutuelle	Cora was interested in what I was saying.	4.36(±1.52)	4.85(±1.25)	4.79(±1.19)	4.42(±1.58)
	Cora was respectful to me and considered to my concerns.	5.21(±1.02)	5.26(±.97)	5.26(±.89)	5.21(±1.09)
Positivité	Cora was warm and caring.	4.21(±1.57)	4.64(±1.30)	4.72(±1.46)	4.13(±1.40)
	Cora was friendly to me.	4.85(±1.22)	5.23(±1.04)	5.17(±.95)	4.91(±1.31)
Rapport	Cora and I established rapport.	3.53(±1.60)***	4.68(±1.30)***	4.23(±1.53)	3.98(±1.60)
	I felt I had no connection with Cora.	2.72(±2.33)	2.11(±2.13)	2.30(±2.21)	2.53(±2.30)

TABLE 1: Questionnaire adapté de [33] pour mesurer la qualité de l'interaction.

Dimensions	Questions	Cora-Mode		User-Mode	
		task-conv	social-conv	button-mode	chat-mode
Confiance décisionnelle	The recipes recommended to me during this interaction matched my preferences.	4.58(±1.71)	4.70(±1.21)	4.91(±1.10)	4.38(±1.74)
Contrôle utilisateur	Cora allowed me to specify and change my preferences during the interaction.	4.53(±1.56)	4.75(±1.40)	4.98(±1.16)	4.30(±1.69)
Intention de réutiliser	I would use Cora to get recipe recommendations in the future.	4.25(±1.89)*	4.98(±1.14)*	4.87(±1.45)	4.36(±1.71)
Effort perçu	I easily found the recipes I was looking for.	4.21(±1.94)	4.75(±1.15)	4.74(±1.44)	4.23(±1.72)
Healthiness	The recipes recommended by Cora were healthy.	4.64(±1.30)	5.04(±1.03)	4.94(±1.14)	4.74(±1.23)
Qualité de recommandation	I was satisfied with the recipes recommended to me.	4.51(±1.68)	4.94(±1.17)	4.85(±1.22)	4.60(±1.66)
Utilité perçue	Cora provided sufficient details about the recipes recommended.	4.09(±1.70)*	4.72(±1.47)*	4.55(±1.43)	4.26(±1.77)
Transparence	Cora explained her reasoning behind the recommendations.	3.34(±1.99)	3.98(±1.86)	3.57(±1.97)	3.75(±1.93)

TABLE 2: Questionnaire adapté de [19] Pour mesurer la qualité du système.

Dimensions	Questions	Cora-Mode		User-Mode	
		task-conv	social-conv	button-mode	chat-mode
Intention de cuisiner	I want to make the recipe recommended to me.	3.98(±1.72)	4.68(±1.24)	4.57(±1.35)	4.09(±1.67)
	I expect to make the recipe recommended to me.	4.40(±1.71)	4.75(±1.11)	4.77(±1.27)	4.38(±1.59)
	It is likely I will make the recipe recommended to me.	4.28(±1.72)	4.72(±1.03)	4.68(±1.30)	4.32(±1.54)
	I intend to make the recipe recommended to me.	4.11(±1.68)	4.45(±1.27)	4.55(±1.37)	4.02(±1.57)
	I will try to make the recipe recommended to me.	4.15(±1.81)	4.53(±1.34)	4.60(±1.47)	4.08(±1.68)

TABLE 3: Questionnaire adapté de [5] pour mesurer l'intention de cuisiner.

## 5 RÉSULTATS

Nous avons collecté 106 interactions sur Amazon Mechanical Turk (N=106) avec un nombre de participants équilibré pour chaque condition. Pour contrôler la qualité des données collectées, chaque participant devait avoir un taux de validation d'au moins 95% et plus de 100 HITs validés. Les participants étaient payés 0.75\$ et passaient en moyenne 6 minutes et 11 secondes sur la tâche (std = 3min59s). Les participants étaient majoritairement des hommes (67%), venant des États-Unis (67%), travaillant à temps plein (80%) et avec un diplôme d'éducation supérieure (89%). Leur âge variait entre 18-29 (24%), 30-49 (65%) ou 50-69 (11%). Les participants cuisinaient au moins une fois par jour (43%) ou plusieurs fois par semaine (44%). La plupart des participants étaient très familiers (55%) ou familiers (41%) avec les assistants conversationnels.

Cora a recommandé 238 recettes au total, avec une moyenne de 2.24 recommandations par interaction (std = 1.69). Le ratio d'avis positifs pour une recommandation est de 0.79 (std = 0.39).

### 5.1 Qualité de l'interaction

Nous effectuons une analyse de variance multivariée factorielle (2x2 Factorial MANOVA) avec Cora-Mode et User-Mode en tant que facteurs intra-sujets. Les mesures dépendantes sont les huit éléments listés en table 1. L'analyse multivariée révèle un effet

principal de Cora-Mode ( $F(1, 102) = 5.088$ ;  $p < .05$ ) sur la qualité perçue de l'interaction. Nous n'avons trouvé aucun effet principal de User-Mode ( $F(1, 102) = 1.214$ ;  $p = .27$ ) sur la qualité de l'interaction, et l'interaction entre les deux variables n'est pas significative ( $F(1; 102) = 3.453$ ;  $p = .06$ ). H1-a est validée, mais pas H2-a.

Pour notre analyse post-hoc, nous étudions les effets univariés de Cora-Mode sur chaque mesure dépendante avec des tests de Student. Nos résultats montrent un effet significatif de Cora-Mode sur l'élément "Cora and I established rapport" ( $F(1, 104) = 16.24$ ;  $p < .001$ ;  $\eta^2 = .14$ ). Cela montre que la version de Cora exprimant des stratégies conversationnelles sociales est bel est bien capable de construire du rapport avec ses participants. Les scores mesurant la qualité de l'interaction sont élevés pour toutes les conditions (moyenne et écart-type pour la qualité de l'interaction :  $m = 4.58$ ,  $std = 1.04$ ). Un résumé des moyennes et des écarts-types pour les huit éléments se trouve dans le tableau 1. Les différences entre les moyennes sont marquées selon leur niveau de signification (\* pour  $p < .05$ , \*\* pour  $p < .005$  et \*\*\* pour  $p < .001$ ).

### 5.2 Qualité du système

Nous effectuons une analyse de variance multivariée factorielle avec Cora-Mode et User-Mode en tant que facteurs intra-sujets. Les mesures dépendantes sont les huit éléments listés en table 2. L'analyse multivariée révèle un effet principal de Cora-Mode ( $F(1,$

102) = 4.325 ;  $p < .05$ ) sur la qualité du système. Nous n'avons trouvé aucun effet principal de User-Mode ( $F(1, 102) = 2.354$  ;  $p = .13$ ) sur la qualité de du système, et l'interaction entre les deux variables n'est pas significative ( $F(1 ; 102) = 1.167$  ;  $p = .28$ ). H1-b est validée, mais pas H2-b.

Comme dans la section précédente, nous étudions les effets univariés de Cora-Mode sur chaque mesure dépendante avec des tests de Student. Nos résultats montrent un effet significatif de Cora-Mode sur les éléments "Intention de réutiliser" ( $F(1, 104) = 5.77$  ;  $p < .05$  ;  $\eta^2 = .05$ ) et "Utilité perçue" ( $F(1, 104) = 4.00$  ;  $p < .05$  ;  $\eta^2 = .04$ ). Cela montre que les participants sont plus favorables à l'idée d'utiliser sur le long-terme un système capable de les engager dans une conversation sociale, et qu'ils perçoivent qu'un tel système donne plus de détails sur les recettes qu'il recommande. Les scores mesurant la qualité du système sont élevés pour toutes les conditions (moyenne et écart-type pour la qualité de l'interaction :  $m = 4.50$  ;  $std = 1.16$ ). Un résumé des moyennes et des écarts-types pour les huit éléments se trouve dans le tableau 2. Les différences entre les moyennes sont marquées selon leur niveau de signification (\* pour  $p < .05$ , \*\* pour  $p < .005$  et \*\*\* pour  $p < .001$ ).

### 5.3 Intention de cuisiner

Nous effectuons une analyse de variance multivariée factorielle avec Cora-Mode et User-Mode en tant que facteurs intra-sujets. Les mesures dépendantes sont les cinq éléments listés en table 3. L'analyse multivariée ne révèle aucun effet principal de Cora-Mode ( $F(1, 102) = 2.660$  ;  $p = .1$ ) ou de User-Mode ( $F(1, 102) = 2.852$  ;  $p = .09$ ) sur l'intention de cuisiner. L'interaction entre les deux variables n'est pas significative ( $F(1 ; 102) = 2.233$  ;  $p = .14$ ). H1-c et H2-c ne sont pas validées. Les scores mesurant l'intention de cuisiner sont élevés pour toutes les conditions (moyenne et écart-type pour la qualité de l'interaction :  $m = 4.41$  ;  $std = 1.39$ ). Un résumé des moyennes et des écarts-types pour les cinq éléments se trouve dans le tableau 3.

### 5.4 Discussion

Les scores élevés obtenus par Cora quelle que soit la condition, combinés au ratio d'avis positifs élevé pour les recettes montre que les participants étaient généralement satisfaits de Cora et de ses recommandations. Nos résultats montrent également que permettre à un système de recommandation de créer du rapport via l'utilisation de stratégies conversationnelles sociales a une influence positive sur la perception de l'utilisateur. Cela est souligné par le fait que la version sociale de Cora a systématiquement obtenu des scores plus élevés que la version tâche, ainsi qu'un écart type plus faible. En d'autres termes, les participants ont préféré la version sociale de Cora, et leurs évaluations étaient plus cohérentes. De plus, non seulement les participants étaient significativement plus favorables à l'idée d'utiliser la version sociale de Cora sur le long-terme, mais ces derniers ont également trouvé que la version sociale de Cora donnait significativement plus de détails que la version tâche, alors que la version sociale donnait seulement son avis sur les recettes. Comme [18], ce résultat souligne l'importance de doter les systèmes de recommandations conversationnels de la capacité d'exprimer leurs propres opinions à propos des choses qu'ils recommandent.

Contrairement à notre intuition initiale, la version chat de Cora a obtenu des scores plus faibles que la version bouton. Nous avons cependant observé deux résultats intéressants. Dans un premier temps, nous notons que les participants ayant interagi avec la version sociale de Cora en langage naturel ont mieux noté la qualité de l'interaction que les participants ayant interagi avec la version sociale de Cora à l'aide de boutons. Cela montre que donner aux utilisateurs la capacité de réciproquer les stratégies conversationnelles du système permet d'améliorer leur expérience. En effet, les participants ayant interagi avec la version sociale de Cora en langage naturel ont jugé l'interaction "amusante et engageante" et que Cora "semblait sympathique et donnait l'impression qu'elle se souciait de l'opinion des gens". Dans un second temps, nous observons que la différence de scores liés à la qualité de l'interaction et à l'intention de cuisiner entre la version bouton et langage naturel était moindre quand Cora utilisait des stratégies conversationnelles sociales. En d'autres termes, les stratégies conversationnelles sociales permettent de mitiger les problèmes liés à la compréhension du langage naturel. Les participants pardonnaient plus facilement les approximations du système dès lors que ce dernier utilisait des stratégies conversationnelles sociales, comme évoqué par un participant : "Cora a ignoré mes préférences alimentaires, mais elle semblait plutôt naturelle par rapport à la plupart des autres chatbots".

Nous avons trouvé une explication plausible aux scores plus faibles obtenus par la version chat de Cora en analysant le contenu des interactions. En effet, certaines des phrases écrites par les utilisateurs n'étaient pas correctement comprises par le système. Nous avons identifié deux types d'erreurs : (1) les phrases incorrectement classifiées par notre module de compréhension du langage naturel, donnant lieu à une recommandation moins précise et (2) les requêtes de l'utilisateur qui n'étaient pas prévues dans notre design, et donc non classifiées, introduisant une frustration chez l'utilisateur. Nous avons observé que ces deux catégories avaient un impact négatif sur les scores obtenus par Cora. Nous avons en outre observé que les participants qui n'étaient pas familiers avec les assistants conversationnels avaient donné des scores significativement plus bas à la version chat de Cora qu'à sa version bouton. Ces participants n'ont peut-être pas conscience des limitations actuelles des assistants conversationnels et préfèrent des interfaces plus conventionnelles qu'ils considèrent plus fiables. Alors que nous nous attendions à ce que les participants trouvent la version bouton de Cora comme plus limitée, seulement un participant a commenté que "certaines réponses me paraissaient restrictives et les réponses toutes faites me donnaient l'impression de ne pas pouvoir dire ce que je voulais".

Pour finir, l'un des commentaires des participants ayant interagi avec la version sociale de Cora met en lumière une notion très intéressante : "C'était bien, mais il y a trop de conversations sans importance à mon avis. Il devrait y avoir deux modes : bavard ou efficace". Cette remarque est consistante avec les travaux présentés par [12] dans lesquels les auteurs classifient les utilisateurs d'un système de recommandation en deux catégories : ceux cherchant activement à construire du rapport, et ceux cherchant uniquement à avoir une bonne recommandation le plus rapidement possible. Construire des systèmes capable d'identifier le type d'utilisateur avec lequel ils interagissent permettrait ainsi d'améliorer grandement l'expérience de ces mêmes utilisateurs.



## 6 CONCLUSION

Dans ce papier, nous avons présenté Cora, un système de recommandation conversationnel capable de recommander des recettes de cuisine correspondant aux habitudes et aux besoins de ses utilisateurs. Cora est capable d'engager ses utilisateurs dans un dialogue social ou simplement orienté tâche, et peut également interagir avec ses utilisateurs à l'aide de boutons ou via langage naturel. Nous avons mené une expérience subjective pour évaluer l'influence du style de conversation de Cora et du mode d'interaction avec l'utilisateur sur la qualité perçue du système et de l'interaction, et sur l'intention de cuisiner. Nos résultats montrent que doter un système de recommandation de stratégies conversationnelles sociales lui permettant de construire du rapport avec ses utilisateurs permet d'améliorer non seulement la qualité perçue de l'interaction, mais également la qualité perçue du système. Nous pensons par ailleurs que ces stratégies conversationnelles sociales sont un moyen de mitiger et de diminuer l'impact négatif d'une erreur de compréhension de la part du système.

L'une des pistes pour améliorer nos travaux serait de remplacer l'API que nous utilisons pour recommander des recettes par notre propre système de recommandation. Nous envisageons actuellement une approche hybride combinant les techniques de factorisation latente avec celles des graphes. Nous pourrions par exemple utiliser des graphes de connaissances [3, 28] pour délivrer des recommandations encore plus personnalisées aux utilisateurs. Nous aimerions également déterminer la meilleure façon de caractériser le profil culinaire d'un utilisateur. En lieu et place d'une moyenne, il serait par exemple intéressant d'optimiser les poids accordés aux habitudes et aux besoins courants lors du processus de recommandation. Nous envisageons également d'introduire un biais afin d'inciter les utilisateurs à choisir des recettes plus saines, et pensons évaluer l'influence d'un tel biais sur l'expérience de l'utilisateur. Enfin, nous envisageons le développement d'une application mobile afin de pouvoir tester l'impact des stratégies conversationnelles sociales sur le long-terme au travers d'une étude longitudinale. Il nous faudra pour cela adapter et améliorer le comportement de l'agent afin d'éviter les répétitions et les incohérences, tout en gardant en mémoire les informations recueillies au cours des interactions.

## RÉFÉRENCES

- [1] Timothy W Bickmore, Laura M Pfeifer Vardoulakis, and Daniel Schulman. 2013. Tinker : a relational agent museum guide. *Autonomous agents and multi-agent systems* 27, 2 (2013), 254–276.
- [2] Justine Cassell and Timothy Bickmore. 2003. Negotiated collusion : Modeling social language and its relationship effects in intelligent agents. *User modeling and user-adapted interaction* 13, 1-2 (2003), 89–132.
- [3] Rose Catherine and William Cohen. 2016. Personalized recommendations using knowledge graphs : A probabilistic logic programming approach. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 325–332.
- [4] Jing Chen, Esther K Papias, and Lawrence W Barsalou. 2016. A core eating network and its modulations underlie diverse eating phenomena. *Brain and cognition* 110 (2016), 20–42.
- [5] Mark Conner, Paul Norman, and Russell Bell. 2002. The theory of planned behavior and healthy eating. *Health psychology* 21, 2 (2002), 194.
- [6] Sally Jo Cunningham and David Bainbridge. 2013. An analysis of cooking queries : Implications for supporting leisure cooking. (2013).
- [7] Aimee L Drolet and Michael W Morris. 2000. Rapport in conflict resolution : Accounting for how face-to-face contact fosters mutual cooperation in mixed-motive conflicts. *Journal of Experimental Social Psychology* 36, 1 (2000), 26–50.
- [8] Jill Freyne and Shlomo Berkovsky. 2010. Intelligent food planning : personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 321–324.
- [9] Jonathan Gratch, Anna Okhmatovskaia, Francois Lamothe, Stacy Marsella, Matthieu Morales, Rick J van der Werf, and Louis-Philippe Morency. 2006. Virtual rapport. In *International Workshop on Intelligent Virtual Agents*. Springer, 14–27.
- [10] Morgan Harvey, Bernd Ludwig, and David Elswiler. 2013. You are what you eat : Learning user tastes for rating prediction. In *International Symposium on String Processing and Information Retrieval*. Springer, 153–164.
- [11] Fritz Heider. 2013. *The psychology of interpersonal relations*. Psychology Press.
- [12] Alankar Jain, Florian Pecune, Yoichi Matsuyama, and Justine Cassell. 2018. A user simulator architecture for socially-aware conversational agents. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. 133–140.
- [13] Michael Jugovac and Dietmar Jannach. 2017. Interacting with recommenders—overview and research directions. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 3 (2017), 10.
- [14] Joseph A Konstan and John Riedl. 2012. Recommender systems : from algorithms to user experience. *User modeling and user-adapted interaction* 22, 1-2 (2012), 101–123.
- [15] SeoYoung Lee and Junho Choi. 2017. Enhancing user experience with conversational agent for movie recommendation : Effects of self-disclosure and reciprocity. *International Journal of Human-Computer Studies* 103 (2017), 95–105.
- [16] Youngme Moon. 2000. Intimate exchanges : Using computers to elicit self-disclosure from consumers. *Journal of consumer research* 26, 4 (2000), 323–339.
- [17] Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2019. An investigation on the user interaction modes of conversational recommender systems for the music domain. *User Modeling and User-Adapted Interaction* (2019), 1–34.
- [18] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A Model of Social Explanations for a Conversational Movie Recommendation System. In *Proceedings of the 7th International Conference on Human-Agent Interaction*. ACM, 135–143.
- [19] Pearl Pu, Li Chen, and Rong Hu. 2011. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 157–164.
- [20] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*. ACM, 117–126.
- [21] Benjamin Scheibehenne, Rainer Greifeneder, and Peter M Todd. 2010. Can there ever be too many options? A meta-analytic review of choice overload. *Journal of consumer research* 37, 3 (2010), 409–425.
- [22] Tanmay Sinha and Justine Cassell. 2015. We click, we align, we learn : Impact of influence and convergence processes on student learning and rapport building. In *Proceedings of the 1st Workshop on Modeling INTERPERSONAL SYNCHRONY AND INFLUENCE*. ACM, 13–20.
- [23] Jan Svennevig. 2000. *Getting acquainted in conversation : A study of initial interactions*. Vol. 64. John Benjamins Publishing.
- [24] Linda Tickle-Degnen and Robert Rosenthal. 1990. The nature of rapport and its nonverbal correlates. *Psychological inquiry* 1, 4 (1990), 285–293.
- [25] Karen Tracy and Nikolas Coupland. 1990. Multiple goals in discourse : An overview of issues. *Journal of Language and Social Psychology* 9, 1–2 (1990), 1–13.
- [26] Christoph Trattner and David Elswiler. 2017. Food recommender systems : important contributions, challenges and future research directions. *arXiv preprint arXiv:1711.02760* (2017).
- [27] David R Traum and Elizabeth A Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational intelligence* 8, 3 (1992), 575–599.
- [28] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN : Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.
- [29] Pontus Wärnestål, Lars Degerstedt, and Arne Jönsson. 2007. Interview and delivery : Dialogue strategies for conversational recommender systems. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*. 199–205.
- [30] Longqi Yang, Cheng-Kang Hsieh, Hongjian Yang, John P Pollak, Nicola Dell, Serge Belongie, Curtis Cole, and Deborah Estrin. 2017. Yum-me : a personalized nutrient-based meal recommender system. *ACM Transactions on Information Systems (TOIS)* 36, 1 (2017), 7.
- [31] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation : System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 177–186.
- [32] Ran Zhao, Alexandros Papangelis, and Justine Cassell. 2014. Towards a dyadic computational model of rapport management for human-virtual agent interaction. In *International Conference on Intelligent Virtual Agents*. Springer, 514–527.
- [33] Ran Zhao, Oscar J Romero, and Alex Rudnicky. 2018. SOGO : a social intelligent negotiation dialogue system. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. ACM, 239–246.