



INSTITUT NATIONAL DES LANGUES ET CIVILISATIONS ORIENTALES

École doctorale n°265 : *Langues, Littératures et Sociétés du monde*

ERTIM

## THÈSE

présentée par

Manying Zhang

soutenance prévue le 16 janvier 2026  
pour obtenir le grade de docteur de l'INALCO  
en Traitement Automatique des Langues

# Agents conversationnels personnalisés et multi-modaux dans le domaine de la mode

### *Thèse dirigée par :*

M. Mathieu Valette	Professeur des universités, INALCO ERTIM
M. Damien Nouvel	Maître de conférences, INALCO ERTIM

### *Rapporteurs :*

M. Kim Gerdes	Professeur des universités, Université Paris-Saclay LISN
M. Nasredine Semmar	Directeur de recherche, CEA LIST

---

### *Membres du jury :*

M <sup>me</sup> Sophie Rosset	Professeur des universités, Université Paris-Saclay LISN
M <sup>me</sup> Béatrice Biancardi	Associate Professor, CESI LINEACT
M. Kim Gerdes	Professeur des universités, Université Paris-Saclay LISN
M. Nasredine Semmar	Directeur de recherche, CEA LIST
M. Mathieu Valette	Professeur des universités, INALCO ERTIM
M. Damien Nouvel	Maître de conférences, INALCO ERTIM



# Résumé

Dans le cadre de cette thèse, nous étudions la conception et le développement d'un agent conversationnel personnalisé et multimodal pour le commerce électronique dans le domaine de la mode. Notre objectif principal est de construire un système capable de comprendre les requêtes complexes des utilisateurs, de recommander des produits pertinents, et de répondre de manière naturelle tout en intégrant des données multimodales (texte, image) et des caractéristiques personnalisées des utilisateurs.

La méthodologie adoptée repose sur trois piliers fondamentaux. Tout d'abord, une stratégie rigoureuse de collecte et génération de données a été mise en place, combinant l'exploitation de corpus existants, la génération automatique via des simulateurs d'utilisateurs et de systèmes, ainsi que des données collectées en ligne. À partir de ces sources, un jeu de données conversationnel dans le domaine de la mode a été constitué, accompagné d'une base de connaissances, comportant des informations produits détaillées, des profils utilisateurs, ainsi que des interactions complexes.

Ensuite, nous avons développé un système modulaire reposant sur plusieurs types de modèles de fondation : des classifieurs, des modèles d'embeddings et des grands modèles de langage (*LLMs*). Ces modèles ont été entraînés ou adaptés individuellement, puis orchestrés dans un pipeline de dialogue structuré, intégrant des modules spécialisés pour la compréhension, la récupération d'information, la génération de réponses, et la personnalisation. Différentes architectures ont été comparées, incluant des approches séquentielles et des modèles de bout-en-bout.

Enfin, nous avons exploré un mécanisme original d'auto-alignement itératif, combinant deux simulateurs : un simulateur d'utilisateur, qui modélise les préférences et les comportements d'achat, et un simulateur de vendeur, qui génère des descriptions de produits à partir de la perspective utilisateur. En confrontant ces deux pôles, le système apprend à générer un langage persuasif pour les produits décrits, et à générer des descriptions en fonction de l'utilité perçue par

## *Résumé*

l'utilisateur simulé.

Les expériences menées montrent que ce système permet une interaction plus personnalisée et pertinente, tout en mettant en lumière les défis méthodologiques liés à la fois à la mise en œuvre d'un système de dialogue, à sa personnalisation et à l'intégration de multimodalité. Ce travail ouvre des perspectives sur le développement d'agents conversationnels dans le cadre d'applications commerciales réelles.

**Mots-clés :** Traitement Automatique des Langues, Agent conversationnel, Intelligence artificielle



# Remerciements

Je tiens tout d’abord à exprimer ma profonde gratitude à mes directeurs de thèse, Mathieu Valette et Damien Nouvel, pour leur soutien constant, leurs conseils avisés et leur patience tout au long de ce parcours doctoral. Mon doctorat a été un long voyage, notamment avec mon retour en Chine en 2021, ce qui a ralenti ma progression. Grâce à la persévérance de tous, nous avons finalement tenu bon — bravo !

Je remercie également les membres du jury, Kim Gerdes, Nasredine Semmar, Sophie Rosset et Béatrice Biancardi, pour le temps précieux qu’ils ont consacré à lire cette thèse couvrant une période très étendue. En effet, les travaux présentés traversent une évolution technique majeure, des méthodes traditionnelles de 2017 aux grands modèles de langage post-2022.

Je souhaite également remercier mes directeurs pour avoir suivi et corrigé cette thèse d’une portée si vaste, ainsi que les évaluateurs annuels du CSID, notamment Patrick Paroubek et Frédéric Landragin, pour leurs encouragements et leurs conseils précieux. Au cours de ces années, le fait de ne pas avoir produit beaucoup d’articles m’a souvent inquiétée ; ils m’ont alors recommandé de me concentrer sur mes travaux existants, de les améliorer et de les publier, ce qui m’a permis de mieux m’intégrer au rythme académique.

Mes remerciements vont également à mes collègues et camarades, pour le partage de ressources, les encouragements mutuels et l’aide apportée à la relecture et à l’amélioration des articles.

Enfin, je remercie ma famille pour son soutien inconditionnel, tant financier que moral et pratique. L’arrivée de notre petit bébé en 2025 a rendu ces années encore plus riches, et sans l’aide de ma famille pour gérer la vie quotidienne, il aurait été difficile de poursuivre ce doctorat.

En regardant en arrière ces sept années, bien que le chemin ait été long et parfois lent, je suis fière d’avoir persévéré. Ce fut une expérience de vie unique et enrichissante.



**Première partie**

**Introduction**



# Chapitre 1

---

## Introduction

### 1.1 Définition et développement des agents conversationnels multimodaux et personnalisé

Les agents conversationnels occupent aujourd’hui une place centrale dans les interactions homme-machine et sont devenus un enjeu de développement du traitement automatique des langues. Ils peuvent être classés en deux grandes catégories : les agents conversationnels à visée générale, comme ceux utilisés pour la conversation libre (par exemple, les chatbots), et les agents orientés-tâche, conçus pour assister les utilisateurs dans l’accomplissement d’objectifs spécifiques, tels que la réservation de billets, la recherche de produits ou la gestion de services clients.

Dans leur forme la plus élémentaire, ces agents reposent sur des systèmes de questions-réponses (*QA*), capables de fournir une réponse préparée à une requête formulée en langage naturel. Cependant, cette approche atteint rapidement ses limites dès lors que l’interaction s’inscrit dans un contexte plus riche ou évolutif. C’est dans ce cadre qu’intervient la distinction entre *QA* et dialogue : un système de dialogue doit non seulement répondre, mais aussi être capable de mémoriser la conversation, maintenir un contexte sur plusieurs tours, adapter son langage et ses suggestions au profil de l’utilisateur et, dans certains cas, apprendre de nouvelles préférences au cours de l’interaction.

Le développement récent des modèles de langage à grande échelle (*Large Language Models*, *LLMs*) a profondément transformé ce domaine. Ces modèles, entraînés sur des corpus massifs, permettent aujourd’hui de concevoir des agents beaucoup plus performants et mieux contrôlés pour le dialogue textuel, et de surcroît capables d’intégrer des modalités multiples (texte, image, voire audio et vidéo) et de fournir des interactions hautement contextuelles et personnali-

sées. C’est ce qu’on désigne désormais sous le terme d’agents conversationnels multimodaux et personnalisés.

La personnalisation permet à l’agent d’ajuster ses réponses selon le profil, les préférences ou l’historique de navigation de l’utilisateur, créant ainsi une expérience d’interaction plus naturelle, fluide et efficace. De plus, la multimodalité permet à l’agent d’interpréter et de générer non seulement du texte, mais aussi des images ou d’autres contenus visuels, ce qui est particulièrement pertinent dans des domaines comme la mode ou le commerce électronique. Par exemple, un utilisateur peut envoyer une image d’un vêtement et demander à l’agent de retrouver des articles similaires, d’en évaluer le style, ou encore de proposer une tenue coordonnée.

Ce double paradigme — multimodalité et personnalisation — redéfinit les contours des agents conversationnels modernes. Dans des domaines fortement visuels, comme celui de la mode, ces capacités ne sont plus simplement accessoires : elles deviennent essentielles pour répondre aux attentes des utilisateurs et améliorer significativement l’engagement et la qualité de l’assistance offerte.

## 1.2 Originalité du projet

### 1.2.1 Contexte du projet

Le présent projet a démarré dans le cadre d’une thèse CIFRE en partenariat avec Southpigalle, une entreprise spécialisée dans les technologies conversationnelles et l’expérience utilisateur. Il vise à concevoir un agent conversationnel capable de comprendre, traiter et générer des interactions complexes autour de la recherche de produits de mode, en tenant compte à la fois des préférences individuelles des utilisateurs et de l’importance de la modalité visuelle propre à ce domaine.

Le besoin fondamental auquel ce projet répond est double : d’une part, les utilisateurs attendent des expériences d’achat en ligne plus personnalisées, naturelles et engageantes ; d’autre part, les marques cherchent à améliorer leur capacité à recommander des produits de manière pertinente et contextuelle, au-delà des simples filtres catégoriels.

Du point de vue industriel, les entreprises proposant des solutions technologiques pour l’e-commerce développent des assistants numériques dans des secteurs variés, avec un intérêt croissant pour les cas d’usage où l’image, le style

et le goût subjectif jouent un rôle central. Le secteur de la mode représente un terrain d'expérimentation privilégié, du fait de sa nature hautement visuelle, émotionnelle, et culturellement marquée.

Sur le plan des technologies, les assistants actuels peinent à exploiter pleinement les modalités disponibles (images, descriptions, interactions utilisateur) et à produire des recommandations cohérentes avec le style visuel ou la personnalité du consommateur. L'introduction d'une approche multimodale et personnalisée permet de dépasser les limites des systèmes actuels en offrant des suggestions sensibles au style, à la silhouette ou encore à la saisonnalité.

Sur le plan scientifique, ce projet se situe à l'intersection de plusieurs avancées récentes : les grands modèles de langage multimodaux, les techniques de fine-tuning, les architectures légères économes en énergie, et les interfaces conversationnelles de nouvelle génération. Il propose ainsi une mise en œuvre concrète et responsable de ces technologies dans un environnement à la fois exigeant du point de vue opérationnel dans le domaine du commerce électronique et contraint selon les moyens disponibles.

### 1.2.2 *Opportunité du projet*

Le contexte technologique est particulièrement favorable à la réalisation de ce projet aujourd'hui. Les progrès rapides dans l'entraînement des grands modèles de langage, la disponibilité accrue de jeux de données multimodaux, et l'émergence d'interfaces de programmation (APIs) facilitant l'intégration de ces modèles dans des systèmes complexes rendent possible ce qui ne l'était pas il y a encore quelques années.

Par ailleurs, la nécessité d'un développement responsable et maîtrisé des technologies d'IA impose de repenser les usages dans une perspective durable : réduire la taille des modèles, limiter leur consommation énergétique, et garantir la transparence des données utilisées. Dans ce contexte, notre projet propose une approche à la fois pragmatique et éthique du développement d'agents intelligents dans un secteur où l'esthétique, la diversité culturelle et les préférences personnelles doivent être prises en compte avec finesse.

### 1.2.3 *Originalité de l'approche*

Notre approche se distingue à plusieurs niveaux. Premièrement, elle repose sur un ensemble de données multilingues, multi-domaines et multimodales, combinant textes, images et annotations stylistiques issues de sources variées, allant des catalogues de produits à des éditoriaux de mode. Cette diversité permet de concevoir un agent capable d'être généralisé à travers les langues, les styles et les cultures.

Deuxièmement, nous nous engageons dans le développement de modèles légers et efficaces, permettant une utilisation sur des infrastructures limitées, réduisant ainsi l'empreinte carbone du déploiement en production. Cette orientation est en cohérence avec une préoccupation croissante pour l'impact environnemental des technologies d'IA.

Troisièmement, nous mettons l'accent sur l'adaptabilité et la robustesse du système, avec des applications potentielles allant du conseil personnalisé à la recherche d'articles similaires, en passant par la stylisation automatique ou la génération d'avis visuels.

Enfin, dans un souci de transparence et de collaboration, nous visons à ouvrir une partie des ressources produites — données, code, modèles — à la communauté scientifique et industrielle. Cette volonté de science ouverte (*open source*) contribue à renforcer la reproductibilité de nos résultats et à stimuler de nouvelles expérimentations dans le domaine.

## 1.3 **Problématique : défis techniques et reconfiguration du dialogue orienté-recommandation à l'ère des LLMs**

Malgré les récents progrès dans les systèmes de dialogue et les systèmes de recommandation, plusieurs obstacles persistent dans la mise en œuvre de véritables agents conversationnels intelligents, capables d'offrir une expérience cohérente, personnalisée, et multimodale dans des domaines complexes comme la mode. Ces obstacles relèvent à la fois de la nature des données, de la fragmentation des tâches, des contraintes computationnelles, et des limites structurelles entre les sous-domaines de l'IA.



### 1.3.1 Des données désordonnées et incomplètes

L'une des premières difficultés rencontrées est la qualité et la structure des données disponibles. Dans le domaine de la mode, les descriptions de produits sont souvent incohérentes, redondantes ou incomplètes ; les images peuvent être mal étiquetées ou peu informatives, et les annotations stylistiques sont rarement normalisées. Par ailleurs, les plateformes d'e-commerce disposent surtout de données issues de comportements implicites (clics, navigation), tandis que les interactions véritablement dialogiques restent limitées et, lorsqu'elles existent, rarement accessibles au public pour des raisons commerciales ou de confidentialité. Cette combinaison rend difficile l'entraînement supervisé classique, et accentue les problèmes de généralisation.

### 1.3.2 Une spécialisation excessive et une évaluation coûteuse

À mesure que les systèmes deviennent plus performants, ils ont également tendance à se spécialiser de manière excessive. Un système conçu pour faire de la classification stylistique ne sait pas nécessairement répondre à des questions subjectives. Un moteur de recommandation est rarement capable de soutenir une conversation contextuelle. Les tâches étant fragmentées, les modèles deviennent nombreux, leur entraînement est complexe, et surtout, leur évaluation devient difficile : il n'existe pas de métrique unique pour juger la qualité d'un système intégré qui comprend, recommande, s'adapte, et dialogue. Cette surcharge rend le cycle de développement long et coûteux.

### 1.3.3 L'apport et les limites des grands modèles de langage

L'arrivée des LLMs a changé la donne. Leur capacité à être généralisés à de nouvelles tâches avec peu d'exemples (*few-shot*), voire sans entraînement spécifique (*zero-shot*), permet de surmonter plusieurs difficultés :

- pour le manque de données, ils permettent la génération automatique (*data augmentation*) de dialogues ou de descriptions ;
- pour les tâches complexes, ils permettent de reformuler des objectifs complexes sous forme d'instructions simples (*instruct learning*) ;
- pour l'évaluation, ils peuvent être utilisés eux-mêmes comme simulateurs de comportement utilisateur pour tester des scénarios variés.

Cependant, les *LLMs* ne sont pas des systèmes conçus spécifiquement pour le dialogue. Leur fonctionnement repose sur une prédiction auto-régressive de texte, sans mémoire structurée ni notion explicite de tour de dialogue ou de but utilisateur. De même, leur lien avec les systèmes de recommandation reste superficiel : ils peuvent expliquer un produit, mais pas forcément optimiser une recommandation sur des critères de pertinence ou de diversité. Il existe donc une double frontière : entre le dialogue et la recommandation, et entre les *LLMs* et les systèmes d'interaction concrets. La première sépare les systèmes de dialogue des systèmes de recommandation. Non seulement leurs finalités divergent — l'un vise la gestion du contexte et des intentions dans un échange langagier, l'autre l'optimisation de la pertinence et de la diversité des suggestions —, mais leurs méthodes sont également distinctes. En recommandation, deux approches dominent : le filtrage basé sur le contenu, qui exploite directement les descriptions ou les images des produits, et le filtrage collaboratif, qui ne requiert pas nécessairement des informations sur les produits, mais seulement des matrices d'interactions utilisateurs-produits. Cette logique diffère fondamentalement de celle des systèmes conversationnels, centrés sur l'interprétation et la génération de langage. La seconde frontière concerne le passage des *LLMs* aux systèmes d'interaction concrets. En effet, un modèle de langage, aussi puissant soit-il, ne constitue pas à lui seul un système opérationnel : un environnement d'usage réel repose sur une combinaison d'interfaces, de modules de navigation, de mécanismes de recherche et de gestion des données. Les *LLMs* offrent donc une composante précieuse, mais doivent être intégrés dans une architecture plus large pour répondre aux contraintes pratiques et aux attentes des utilisateurs.

#### 1.3.4 *Vers une architecture hybride : arbitrages et contributions*

Dans ce contexte, notre approche vise à tirer parti de la puissance des *LLMs* tout en réintroduisant des structures modulaires permettant une meilleure efficacité, une adaptation plus fine aux cas d'usage, et une maîtrise des ressources. Elle repose sur plusieurs arbitrages fondamentaux :

- entre l'universalité des *LLMs* et la spécificité des modules métiers,
- entre la précision des recommandations et la capacité d'exploration,
- entre la qualité algorithmique et l'ergonomie de l'interface utilisateur,
- entre l'expressivité du langage naturel et les contraintes computationnelles

réelles.

En intégrant les *LLMs* comme noyau sémantique — capables de reformuler les besoins, d'enrichir les données, et de simuler des comportements — et en les combinant à une architecture modulaire de recommandation, notre système offre une nouvelle manière d'aborder les agents conversationnels orientés-tâche. Il ne s'agit plus de résoudre une tâche précise avec un modèle dédié, mais d'orchestrer des compétences générales autour d'un dialogue personnalisé, contextualisé, et enrichi par la possibilité de lui fournir des images. Ce fonctionnement n'est toutefois possible que grâce à l'exploitation de ressources adaptées — données spécialisées, infrastructures de calcul, et méthodes d'affinage — sans lesquelles les *LLMs* ne pourraient pas atteindre une efficacité réelle.

### 1.3.5 Des exigences computationnelles croissantes

De nombreuses expériences dans la communauté du TAL montrent que les performances des modèles s'améliorent généralement avec l'augmentation de la quantité de données, du nombre de paramètres et de la capacité de calcul. Ce constat est souvent formulé, dans les recherches sur les grands modèles de langage (*LLM*), sous le nom de « loi d'échelle » (*scale law*). Toutefois, cette tendance s'accompagne du problème du passage à l'échelle : plus un modèle devient volumineux, plus il devient difficile de maintenir son efficacité, sa stabilité et sa durabilité, que ce soit sur le plan énergétique, éthique ou industriel. Cette tension impose également des choix stratégiques : que faut-il optimiser ? La compréhension, la recommandation, l'interaction ? Faut-il entraîner un modèle par tâche ou concevoir une architecture modulaire ?



## Chapitre 2

---

### État de l’art et recherches liées

#### 2.1 TAL et les agents conversationnels

Selon les applications, les agents conversationnels sont généralement divisés en deux catégories : ceux orientés-tâches (*TOD*, *Task-Oriented Dialogue*) et ceux à domaine ouvert (*ODD*, *Out-of-Domain*). Les agents *TOD* résolvent des problèmes spécifiques dans un domaine donné, tels que la réservation de billets de cinéma, la réservation de tables de restaurant, etc. Au lieu de se concentrer sur l’accomplissement d’une tâche, les agents *ODD* cherchent à discuter avec les utilisateurs sans restriction de tâche et de domaine.

D’après l’avis du travail de NI et al. (2021), les agents conversationnels, qu’ils soient *TOD* ou *ODD*, peuvent être vus comme une correspondance  $\varphi$  entre le message de l’utilisateur  $U = \{u^{(1)}, u^{(2)}, \dots, u^{(i)}\}$  et la réponse de l’agent  $R = \{r^{(1)}, r^{(2)}, \dots, r^{(j)}\}$  :

$$R = \varphi(U) \quad (2.1)$$

où  $u^{(i)}$  et  $r^{(j)}$  désignent respectivement le  $i^e$  token du message de l’utilisateur et le  $j^e$  token de la réponse de l’agent. La simplicité et la clarté de la formulation jouent un rôle crucial dans la compréhension de la logique des dialogues à une seule étape. Toutefois, lorsqu’il s’agit de dialogues à plusieurs étapes, l’historique du dialogue  $H$  devient indispensable. Cet historique  $H$  englobe toutes les messages antérieurs, représenté par  $H = \{(U_1, R_1), (U_2, R_2), \dots, (U_n, R_n)\}$ , où  $U_i$  et  $R_i$  désignent respectivement le message du  $i^e$  tour de l’utilisateur et de l’agent, avec  $n = t - 1$  sachant que  $t$  est le tour présent. Ainsi, la représentation mathématique concise des systèmes de dialogue sera plutôt :

$$R = \varphi(U, H) \quad (2.2)$$

. Dans de nombreux scénarios tels que le nôtre, cette correspondance prend également en compte une source de connaissances externes ou une base de données comme un catalogue  $K$  en tant qu'entrée :

$$R = \varphi(U, H, K) \quad (2.3)$$

On peut encore étendre cette formulation pour inclure les éléments supplémentaires, par exemple, les personnalités de l'utilisateur ou de l'agent, que nous notons  $P$  :

$$R = \varphi(U, H, K, P) \quad (2.4)$$

Les formules ci-dessus résument simplement quelques types courants d'entrées pour les systèmes de dialogue. En plus des messages de l'utilisateur  $U$  comme base, les autres éléments tels que  $H$ ,  $K$ ,  $P$  peuvent être des données facultatives dans différents contextes. En ce qui concerne la forme des entrées, nous considérons également la possibilité d'une multimodalité, incluant le texte, l'audio, le visuel, leurs combinaisons, l'interaction utilisateur avec des boutons, et bien d'autres.

À propos de la fonction de correspondance  $\varphi$ , il existe plusieurs méthodes, qui peuvent utiliser des systèmes à base de règles, ou des modèles d'apprentissage profond, par un processus en chaîne de traitement ou de bout en bout. Les agents conversationnels *TOD* traditionnels sont organisés selon une structure en chaîne de traitement et comprennent quatre modules fonctionnels : la compréhension du langage naturel (*NLU*, *Natural Language Understanding*), le suivi de l'état du dialogue (*DST*, *Dialogue State Tracking*), l'apprentissage des stratégies (*PL*, *Policy Learning*) et la génération de langage naturel (*NLG*, *Natural Language Generation*). De l'autre part, l'approche de bout en bout est largement explorée dans la littérature récente, notamment pour sa capacité potentielle à simplifier la structure du système et réduire les erreurs de propagation induites par les chaînes de traitements. Ces travaux seront détaillés dans la section 2.3. La section 2.2 résume de manière exhaustive les ensembles de données couramment utilisés pour les agents conversationnels, en particulier les agents *TOD*. Les sections 2.5 et 2.6 se concentrent respectivement sur la personnalisation et la multimodalité dans les agents conversationnels.

## 2.2 Jeux de données

Étant donné notre scénario d’achat d’articles de mode, nous considérons notre tâche comme étant la recommandation de produits de mode par un agent conversationnel orienté-tâche, qui diffère beaucoup d’autres tâches couramment rencontrées telles que le réglage de l’alarme ou la réservation de restaurant, puisque l’utilisateur peut ne pas avoir une idée claire de ce qu’il souhaite acheter. Pour cette raison, nous abordons également certains travaux sur les systèmes de recommandation conversationnels, en dehors des ensembles de données couramment utilisés pour les agents *TOD*. Pour ces agents conversationnels, les données de dialogue utiles à l’élaboration du système peuvent être obtenues par collecte ou par génération. Pour collecter les dialogues, nous trouvons deux méthodes majoritaires. L’une consiste à recruter des travailleurs humains, par exemple, via des plateformes de « crowdsourcing » pour produire des dialogues pour une tâche donnée. L’autre consiste à collecter des dialogues lors de véritables accomplissements de tâches, que ce soit entre deux êtres-humains ou entre humain et machine. De l’autre côté, pour générer les dialogues, les simulateurs de l’utilisateur et les APIs (Interface de Programmation Applicative) spécifiques simulant les agents sont souvent pris en compte. Ces ensembles de données sont revus et catégorisés, avec une synthèse présentée dans le tableau 2.1. Une version plus détaillée figure dans l’annexe A.1.

TAB. 2.1 : Jeux de données *TOD*

Titre	Domaines	Objectif	Méthode	Citation
JDDC	E-commerce	<i>TOD</i>	Collection des dialogues réels	M. CHEN et al. 2019
PersonaChat	Bavardage	<i>ODD</i> , Personnali- sation	Collecte via la méthode WOZ (personas et dialogues)	Saizheng ZHANG et al. 2018
PersonaChat Gen	Bavardage	<i>ODD</i> , Personnali- sation	Simulation des personas et des conversations par GPT <sub>3</sub>	Y.-J. LEE, LIM et al. 2022

SalesBot	Bavardage, cinéma	ODD, TOD	Simulation puis reformulation par les modèles de langage larges	CHIU et al. 2022
MMD	E-commerce de la mode	TOD, Multimodal	Collection des dialogues par crowdsourcing	A. SAHA, KHAPRA et SANKARANARAYANAN 2017
SIMMC	E-commerce de la mode et les meubles	TOD, Multimodal	Collection des dialogues par crowdsourcing	CROOK et al. 2019
MMConv	Hôtel, centre commercial, etc	TOD, Multimodal	Collection des dialogues par crowdsourcing	L. LIAO et al. 2021
MMDialog	Domaine général	ODD, Multimodal	Collection des dialogues réels	FENG et al. 2022

## 2.3 Architecture et modèle d'agents conversationnels

Les agents conversationnels traditionnels sont principalement basés sur des règles (ARORA, K. BATRA et S. SINGH 2013) ou des systèmes d'apprentissage automatique non neuronaux. Les agents basés sur des règles sont faciles à mettre en œuvre et peuvent fournir des réponses scriptées, ce qui a contribué à leur popularité dans les produits industriels. Cependant, les messages émis par ces systèmes sont prédéterminés, ce qui limite les applications des systèmes de dialogue à certains scénarios. Les systèmes d'apprentissage automatique non neuronaux gèrent certaines tâches grâce à des modèles statistiques simples. Ces systèmes sont plus flexibles par rapport aux précédents mais sont également limités en termes de scénarios d'application et de diversité des réponses. La plupart des agents conversationnels récents sont basés sur l'apprentissage profond (systèmes neuronaux), pour lesquels les performances des systèmes sont largement



améliorées (Hongshen CHEN et al. 2017).

Dans cette partie, nous allons passer en revue certains des modèles neuronaux et des mécanismes largement utilisés dans les agents conversationnels, en particulier dans les systèmes *TOD*. Nous approfondirons ensuite les architectures couramment vues dans les systèmes *TOD*. Ensuite, nous aborderons les travaux existants concernant la personnalisation, l'ancrage des connaissances et la multimodalité dans les tâches de dialogue. Enfin, nous aborderons brièvement les mises en œuvres des agents dans l'industrie de la mode.

### 2.3.1 Modèles neuronaux dans les agents conversationnels

Dans cette section, nous introduisons des modèles neuronaux populaires pour réaliser les agents conversationnels et les sous-tâches connexes. Nous discutons également des applications de ces modèles ou de leurs variantes dans les récentes recherches sur les systèmes de dialogue. Les modèles discutés comprennent : les modèles d'encodeur-décodeur, les mécanismes d'attention (*Attention Networks*), les Transformers, et les grands modèles de langue (*LLM*).

#### 2.3.1.1 Encodeur-Décodeur

SUTSKEVER, VINYALS et Q. V. LE (2014) a proposé le modèle de séquence-à-séquence pour résoudre les tâches de traduction automatique. Ce modèle vise à faire une correspondance entre une séquence d'entrée et une séquence de sortie en utilisant d'abord un encodeur pour transformer la séquence d'entrée en un vecteur intermédiaire, puis un décodeur génère la sortie basée sur le vecteur intermédiaire et l'historique généré par le décodeur, décrit par les équations suivantes :

$$Encoder : h_t = E(h_{t-1}, x_t) Decoder : y_t = D(h_t, y_{t-1}) \quad (2.5)$$

Où  $t$  est l'étape temporelle,  $h$  est le vecteur caché et  $x$  et  $y$  sont les vecteurs d'entrée et de sortie.  $E$  et  $D$  sont les cellules récurrentes (par ex. *RNN*, *LSTM* ou *GRU*) utilisées respectivement par l'encodeur et le décodeur. Le dernier état caché de l'encodeur  $h_t$  est le vecteur intermédiaire, qui est généralement utilisé pour initialiser le premier état caché du décodeur. Lors de l'encodage, chaque état caché est déterminé par l'état caché de l'étape temporelle précédente et l'entrée à l'étape temporelle actuelle, tandis qu'à l'étape de décodage, chaque

état caché est déterminé par l'état caché actuel et la sortie de l'étape temporelle précédente.

Ce modèle est puissant car il n'est pas limité à des entrées et des sorties de longueur fixe. Au lieu de cela, les longueurs des séquences d'entrée et de sortie peuvent être différentes. Les *RNN* (Réseaux Neuronaux Récurrents) sont souvent utilisés comme encodeur pour encoder le contexte du dialogue, l'état du dialogue, les entrées de la base de connaissances ou les balises de domaine dans les systèmes *TOD* (W. CHEN et al. 2019 ; MOON et al. 2019a ; J. WU, X. E. WANG et W. Y. WANG 2019) et pour encoder l'historique du dialogue (X. CHEN, MENG et al. 2020 ; W. DU et BLACK 2019 ; SANKAR et al. 2019) ou ensemble avec les réponses de candidat (FELDMAN et EL-YANIV 2019 ; TANG et al. 2019 ; Q. ZHU et al. 2018) pour les systèmes *ODD*. Dans les systèmes fondés sur les connaissances (X. CHEN, MENG et al. 2020 ; MAJUMDER, Shuyang LI et al. 2020 ; SHUSTER, JU et al. 2019), les *RNNs* encodent aussi les sources de connaissances externes (par exemple, le contexte, la personnalité, le sujet, etc.). Tandis que pour la partie du décodeur, les *RNNs* sont aussi utilisés pour décoder l'état caché des séquences (Y. HUANG et al. 2020 ; X. LIN et al. 2019 ; Z. SONG et al. 2019). De plus, certains travaux ont utilisé les *RNNs* pour entraîner aussi les modèles de plongement (*embedding*) de dialogue afin d'améliorer les performances du modèle (X. CHEN, J. XU et B. XU 2019 ; C. LIU et al. 2019 ; MA et al. 2020 ; MEHRI et al. 2019). En profitant de l'information spécifique aux tâches, ces modèles ont souvent surpassé les modèles de représentation contextuelle de l'état de l'art (par exemple, *BERT*, *ELMo* et *GPT*) lorsque ces derniers n'étaient pas affinés.

### 2.3.1.2 Mécanisme d'Attention

Comme décrit dans la section 2.3.1.1 et dans l'équation 2.5, un modèle de décodeur classique décode le prochain token en fonction de l'état caché actuel et du vecteur de sortie de la dernière étape de temps. L'équation 2.5 peut être reformulé comme suit :

$$P(y_t|y_1, \dots, y_{t-1}, x) = g(y_{t-1}, h_t) \quad (2.6)$$

où  $t$  représente la  $t$ -ème étape de temps ;  $P(y_t|y_1, \dots, y_{t-1}, x)$  représente la probabilité de générer le token  $y_t$  donné les tokens précédents et l'entrée  $x$ ,  $h_t$  est l'état caché à l'instant  $t$ , et  $g$  est une fonction de projection (par exemple

une couche linéaire suivie d'une *softmax*). Cependant, un tel schéma de décodage présente des limites lorsque la séquence d'entrée est longue. Les *RNNs* peuvent difficilement encoder toutes les informations dans un vecteur caché de longueur fixe (CHO et al. 2014). De plus, en raison de la capacité d'expression limitée d'un vecteur caché de longueur fixe, les performances du schéma de décodage de la formule 2.6 dépendent largement des premières étapes du décodage, et si le décodeur ne démarre pas correctement, toute la séquence sera négativement affectée en raison d'un phénomène de propagation d'erreur.

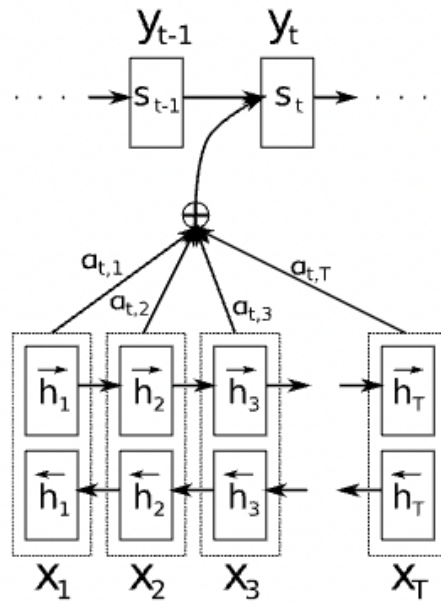


FIG. 2.1 : La structure de modèle d'attention (BAHDANAU, CHO et BENGIO 2014).

Dans ce contexte, BAH DANAU, CHO et BENGIO (2014) ont proposé le mécanisme d'attention dans la tâche de traduction automatique. Ils ont décrit cette méthode comme « alignement et traduction conjoints », qui implémente le modèle de traduction séquence à séquence comme un modèle encodeur-décodeur avec attention. Comme montré dans la figure 2.1, à l'étape de décodage, chaque état de décodage prendrait en compte les parties de la phrase source encodée qui sont corrélées, plutôt que de dépendre uniquement du token de sortie précédent

immédiat. La distribution de probabilités de sortie peut être décrite comme suit :

$$P(y_t|y_1, \dots, y_{t-1}, x) = g(y_{t-1}, s_t, c_t) \quad (2.7)$$

où  $y_t$  est le token de sortie,  $s_t$  est l'état caché du décodeur et  $c_t$  est la phrase source pondérée calculée depuis les états cachés de l'encodeur  $h$  :

$$s_t = f(s_{t-1}, y_{t-1}, c_t) \quad (2.8)$$

$$c_t = \sum_{j=1}^{T_x} a_{tj} h_j \quad (2.9)$$

où  $a_{tj}$  représente le score de poids normalisé, autrement dit l'attention entre les états cachés du  $j$ -ème token du décodeur et du  $t$ -ème token de l'encodeur :

$$a_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T_x} \exp(e_{tk})} \quad (2.10)$$

### 2.3.1.3 Transformer

Avant les transformers, la plupart des travaux combinaient l'attention avec des unités récurrentes. Ces modèles récurrents conditionnent chaque état caché sur l'état caché précédent et l'entrée actuelle, et ils sont flexibles en termes de longueur de séquence. Cependant, en raison de leur nature séquentielle, les modèles récurrents ne peuvent pas être entraînés en parallèle, ce qui compromet sérieusement leur calculabilité sur de grands volumes de données. VASWANI et al. (2017) ont proposé le transformer, qui utilise entièrement des mécanismes d'attention sans aucune unité récurrente et déploie davantage de parallélisme pour accélérer l'entraînement. Il applique l'auto-attention et l'attention encodeur-décodeur pour capturer respectivement les dépendances locales et globales.

Comme montré dans la figure 2.2, le transformer se compose d'un encodeur et d'un décodeur. L'encodeur calcule pour la séquence d'entrée  $(x_1, \dots, x_n)$  les états cachés continus  $(h_1, \dots, h_n)$ . Le décodeur génère ensuite la séquence de sortie  $(y_1, \dots, y_n)$  en fonction des états cachés de l'encodeur. Le modèle de probabilité du Transformer est sous la même forme que celui du modèle séquence à séquence classique introduit dans la section 2.3.1.1. VASWANI et al. (2017) ont empilé 6 couches d'encodeurs identiques et 6 couches de décodeurs identiques. Une

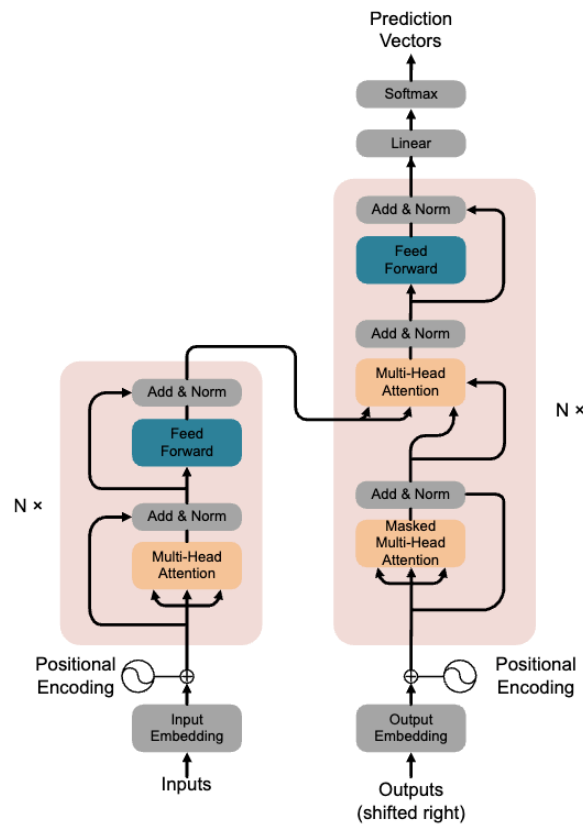


FIG. 2.2 : La structure de transformer (VASWANI et al. 2017).

couche d'encodeur se compose d'un composant d'attention multi-têtes et d'un réseau *feed-forward* simple, tous deux avec une connexion résiduelle permettant de faciliter la propagation du gradient. La structure d'une couche de décodeur est presque identique à celle d'une couche d'encodeur, à l'exception d'une couche d'attention supplémentaire entre les états cachés du décodeur de l'étape temporelle actuelle et les vecteurs de sortie de l'encodeur. L'entrée du décodeur est partiellement masquée pour s'assurer que chaque prédiction est basée sur les tokens précédents, évitant ainsi de prédire en tenant compte d'informations futures. Les deux entrées de l'encodeur et du décodeur utilisent un mécanisme d'encodage de position pour fournir des informations sur la position de chaque élément dans la séquence.

Le Transformer a révolutionné le domaine du traitement du langage naturel en permettant des entraînements parallèles plus efficaces, ce qui a permis d'accélérer

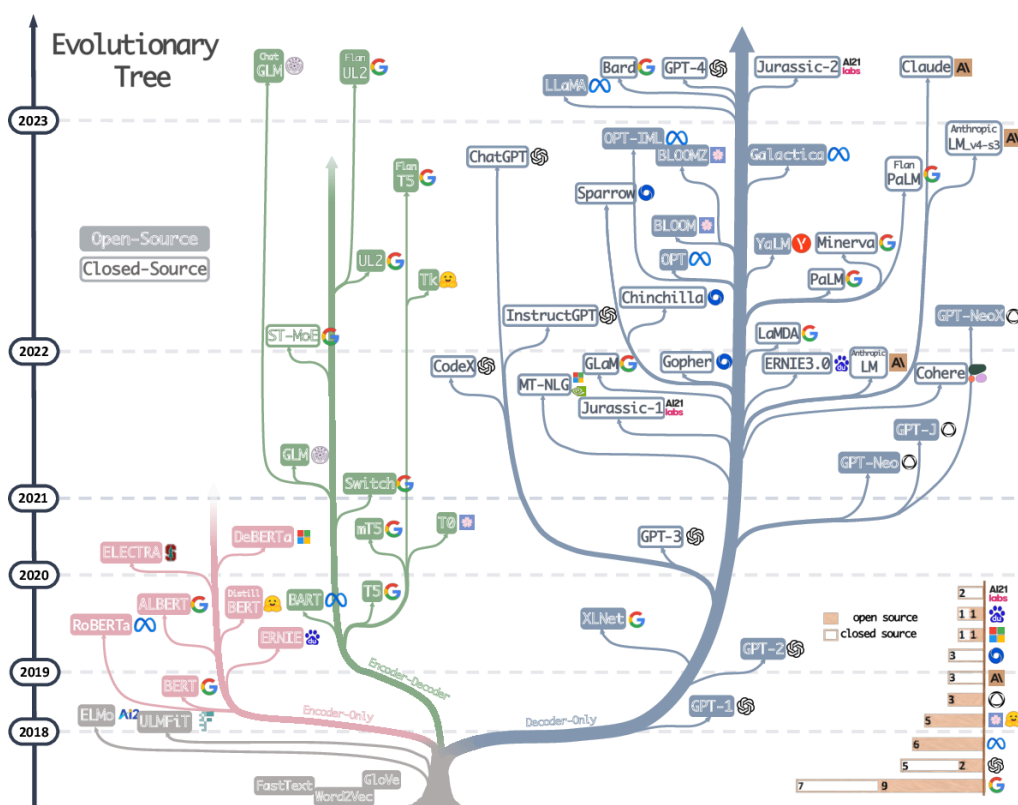


FIG. 2.3 : Résumé de grands modèles de langage avant 2023 (J. YANG et al. 2023).

considérablement les temps d'entraînement. En remplaçant les unités récurrentes par des mécanismes d'attention, le Transformer a également montré de meilleures performances dans de nombreuses tâches de traitement du langage naturel. L'utilisation de l'auto-attention permet au modèle de capturer les relations entre les mots dans une séquence de manière globale, tandis que l'attention encodeur-décodeur aide à modéliser les dépendances entre les séquences source et cible lors des tâches de traduction et autres tâches de séquence à séquence.

#### 2.3.1.4 Grands modèles de langue (LLM)

Nous voyons que la modélisation du langage a été largement étudiée pour la compréhension et la génération de langage au cours des deux dernières décennies, passant des modèles statistiques aux modèles neuronaux comme des *Transformers*. Récemment, des modèles de langue pré-entraînés (PLMs) ont été

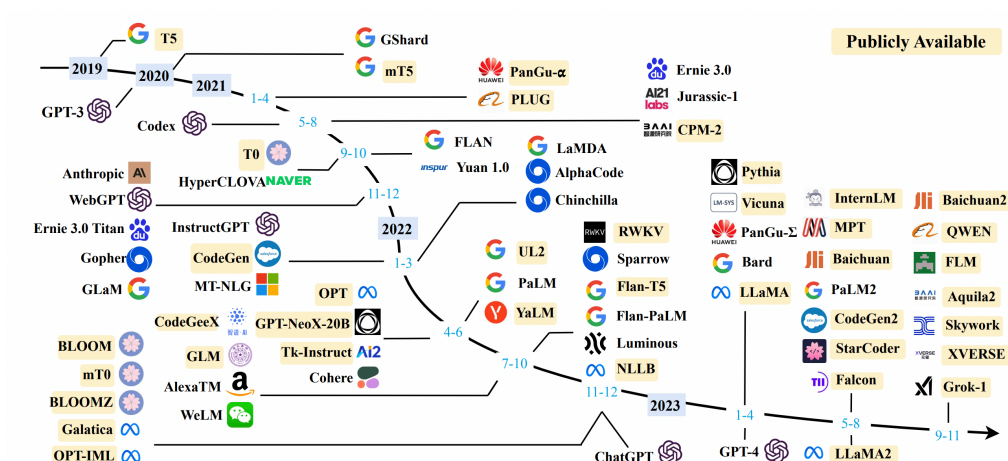


FIG. 2.4 : Les LLMs duquels le nombre de paramètres surpasse 10 milliards, nés 2023, avec ligne du temps (W. X. ZHAO et al. 2023 ).

proposés en appliquant des modèles *Transformer* sur des corpus à grande échelle, démontrant de très bonnes capacités à résoudre diverses tâches du traitement de langue naturelle. Étant donné que les chercheurs ont constaté, conformément à la « loi d'échelle », que l'augmentation de la taille du modèle, du volume de données et des ressources de calcul conduit généralement à une amélioration systématique des performances, ils ont exploré davantage cet effet en augmentant l'échelle des paramètres à une taille encore plus grande. De manière intéressante, lorsque l'échelle des paramètres dépasse un certain niveau, ces modèles de langue n'améliorent pas seulement considérablement les performances des tâches cibles, mais présentent également certaines capacités spéciales et généralisées (par exemple, l'apprentissage en contexte) qui ne sont pas présentes dans les modèles de langue à petite échelle (par exemple, *BERT*). Pour discriminer les modèles de langue selon différentes échelles de paramètres, la communauté de recherche a inventé le terme "grands modèles de langage" (*LLM*) pour les *PLMs* de taille significative (par exemple, contenant des dizaines ou des centaines de milliards de paramètres). La figure 2.3 présente un résumé des LLMs et de leurs tailles, et la figure 2.5 illustre la tendance d'augmentation du nombre de paramètres des LLMs ces dernières années.

Au-delà de la simple modélisation de la probabilité des séquences, les *LLMs*

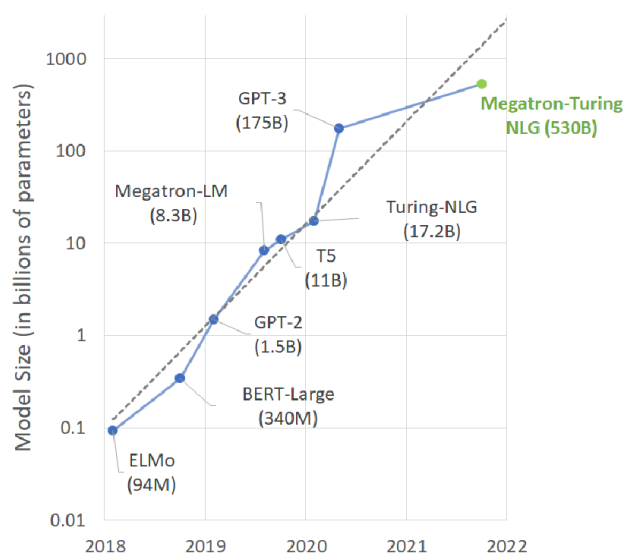


FIG. 2.5 : Augmentation du nombre de paramètres des modèles au cours du temps (S. SMITH et al. 2022).

récents ont montré leur capacité à interagir de manière plus flexible avec des instructions humaines. L'introduction du *prompting* (BROWN et al. 2020) et de l'*instruction tuning* (MISHRA et al. 2021 ; OUYANG et al. 2022) permet aux modèles de suivre des consignes explicites formulées en langage naturel, ouvrant la voie au *zero-shot* et au *few-shot learning*.

Parallèlement, des méthodes de raisonnement explicite comme le *Chain-of-Thought* (CoT, WEI et al. 2022) ont été proposées pour guider le modèle dans la production d'étapes intermédiaires, améliorant ses performances sur les tâches complexes de raisonnement symbolique ou arithmétique.

Pour enrichir la base de connaissances interne des LLMs, souvent limitée aux données d'entraînement, des approches hybrides telles que la *Retrieval-Augmented Generation* (RAG, P. LEWIS et al. 2020) ont été développées. Elles combinent génération et recherche documentaire, en permettant au modèle d'accéder à des sources externes pour produire des réponses plus précises et actualisées.

De nombreux modèles emblématiques illustrent ces avancées. GPT-3 et GPT-4 (BROWN et al. 2020 ; OPENAI 2023) ont marqué un tournant dans la génération de texte généraliste grâce à un pré-entraînement sur des corpus massifs issus du web, de livres et de code. Les modèles PaLM et Gemini de Google (CHOWDHERY et al. 2022 ; GEMINI TEAM GOOGLE 2023) reposent sur une combinaison de données



web et scientifiques multilingues, optimisées pour des tâches multimodales. Les modèles de la série *LLaMA* et ses dérivés (*Alpaca*, *Vicuna*, *Falcon*, etc.) (META AI 2024 ; TAORI et al. 2023 ; TOUVRON, LAVRIL et al. 2023 ; TOUVRON, MARTIN et al. 2023) ont permis la démocratisation de l'accès aux *LLMs* open-source, en mettant l'accent sur la reproductibilité et l'adaptation par affinage d'instructions synthétiques. Enfin, la famille *Qwen* (BAI et al. 2023) illustre une approche multilingue issue de la recherche chinoise, combinant apprentissage auto-régressif et masqué pour une meilleure efficacité d'entraînement. Récemment, plusieurs modèles de taille intermédiaire ont également montré des performances remarquables tout en étant plus légers que les très grands modèles de centaines de milliards de paramètres. Parmi eux, *Mistral-7B* (A. Q. JIANG, SABLAYROLLES, MENSCH et al. 2023 ; A. Q. JIANG, SABLAYROLLES, ROUX et al. 2024) se distingue par ses 7 milliards de paramètres, ce qui est nettement inférieur aux modèles massifs comme GPT-4 ou PaLM, mais il parvient à atteindre des performances comparables sur plusieurs benchmarks de compréhension et de génération.

Ces avancées techniques et ces évolutions d'architectures confirment que les *LLMs* modernes ne se limitent plus à la modélisation du langage, mais sont capables de compréhension, de raisonnement et d'interaction contextualisée. Leur performance dépend à la fois de la qualité des données de pré-entraînement, des méthodes d'affinage (*fine-tuning* et *instruction tuning*), et des mécanismes d'intégration de connaissances externes tels que RAG.

### 2.3.2 Architectures pour les agents conversationnels TOD

Dans cette partie, nous allons comparer les systèmes *TOD* dans deux différentes architectures : modulaire et de bout en bout. Un agent conversationnel *TOD* nécessite des contraintes de réponse plus strictes car il vise à gérer avec précision les messages de l'utilisateur. Par conséquent, des méthodes modulaires ont été proposées pour générer des réponses de manière plus contrôlée. L'architecture d'un système basé sur des modules se compose de quatre modules :

- Compréhension de langue naturelle, *NLU*
- Suivi des états du dialogue, *DST*
- Apprentissage des stratégies, *PL*
- Génération de langue naturelle, *NLG*

### 2.3.2.1 Module de compréhension de langue naturelle (Natural Language Understanding - *NLU*)

Ce module est responsable de comprendre le message de l'utilisateur, d'extraire les intentions, les emplacements (*slot*) et les domaines concernés pour la tâche spécifique, autrement dit, le module *NLU* gère trois tâches : la classification de domaine, la détection d'intention et l'extraction des *slots*. La classification de domaine et la détection d'intention sont des problèmes de classification, qui utilisent des classifieurs pour prédire une correspondance entre la séquence de langage en entrée et un ensemble prédéfini d'étiquettes, tandis que l'extraction des *slots* est un problème d'étiquetage. Les *RNNs* et les *CNNs* sont souvent utilisés pour encoder les énoncés et les historiques de dialogue avant la classification (HASHEMI 2016 ; J. Y. LEE et DERNONCOURT 2016 ; RAVURI et STOLCKE 2015, 2016). Plus récemment, C.-S. WU, HOI et al. (2020) ont pré-entraîné Task-Oriented Dialogue BERT (*TOD-BERT*) et ont considérablement amélioré la précision dans la sous-tâche de détection d'intention. Le modèle proposé a également démontré une grande capacité d'apprentissage de *few-shot* et a pu atténuer efficacement le problème d'insuffisance de données dans un domaine spécifique. Pour l'extraction des *slots*, les *RNNs*, y compris les *LSTM* sont plus populaires (MESNIL, DAUPHIN et al. 2015 ; MESNIL, Xiaodong HE et al. 2013 ; YAO, Baolin PENG et al. 2014 ; YAO, ZWEIG et al. 2013). Certaines études choisissent de combiner la classification de domaine, la détection d'intention et le remplissage des *slots* dans un cadre d'apprentissage multi-tâches afin d'optimiser conjointement l'espace latent partagé. HAKKANI-TÜR et al. (2016) ont utilisé une architecture bi-directionnelle *RNN-LSTM* pour effectuer conjointement les trois tâches. B. LIU et LANE (2016) ont augmenté le modèle traditionnel d'encodeur-décodeur *RNN* avec un mécanisme d'attention pour gérer la détection d'intention et l'extraction des *slots*. L'apprentissage multi-tâches des trois sous-tâches de *NLU* a facilité l'entraînement en *zero-shot* ou *few-shot* lors du transfert vers un nouveau domaine (BAPNA et al. 2017 ; S. LEE et JHA 2018). Une autre approche proposée par S.-Y. SU, C.-W. HUANG et Y.-N. CHEN (2019) repose sur l'idée que *NLU* et *NLG* étaient des processus inverses. Ainsi, le modèle pourrait être entraîné avec un cadre d'apprentissage à double supervision.

### 2.3.2.2 Module de suivi des états du dialogue (Dialogue State Tracking - DST)

Ce module prend en charge de l'utilisateur et les détails connexes à chaque tour en fonction de l'ensemble de l'historique du dialogue pour fournir les informations sur lesquelles le module d'apprentissage de la politique (prochain module) décide de l'action de l'agent à réaliser. La différence entre le module *NLU* et le *DST* est que le *NLU* classe l'intention et étiquette les *slots* à partir de l'énoncé actuel de l'utilisateur, tandis que le *DST* prend en compte l'historique complet du dialogue afin de mettre à jour une représentation structurée de l'état du dialogue, comprenant l'ensemble des intentions et des *slots* identifiés. Certains systèmes de dialogue prennent la sortie du module *NLU* comme entrée du module *DST* (M. HENDERSON, THOMSON et J. WILLIAMS 2014a,b ; J. WILLIAMS, RAUX et al. 2013), tandis que d'autres ont directement utilisé les messages bruts de l'utilisateur pour suivre l'état (HU et al. 2020 ; Sungdong KIM et al. 2019 ; Ye WANG, GUO et S. ZHU 2020). Nous mentionnons les ateliers de défis de suivi des états du dialogue (Dialogue State Tracking Challenges - *DSTCs*), une série de défis populaires en *DST*, qui fournissent des ensembles de données de référence, des cadres d'évaluation standard et des environnements de test pour la recherche (M. HENDERSON, THOMSON et J. WILLIAMS 2014a,b ; J. WILLIAMS, RAUX et al. 2013).

Les premiers travaux utilisaient des règles artisanales ou des méthodes statistiques pour résoudre les tâches *DST*. Bien qu'elles soient largement utilisées dans les systèmes de dialogue industriels, les méthodes *DST* basées sur des règles (GODDEAU et al. 1996) ont de nombreuses restrictions, telles qu'une généralisation limitée, un taux d'erreur élevé, une faible adaptabilité au domaine, etc. (J. WILLIAMS 2014). Les méthodes statistiques (Seokhwan KIM et BANCHS 2014 ; S. LEE 2013 ; S. LEE et ESKÉNAZI 2013 ; REN et al. 2013 ; J. WILLIAMS 2014 ; Jason WILLIAMS 2013) souffrent également de bruits et d'ambiguïtés (THOMSON, SCHATZMANN et YOUNG 2008 ; YOUNG et al. 2010).

Récemment, de nombreux modèles neuronaux ont émergé. Les suiveurs neuronaux présentent plusieurs avantages. En général, ils sont catégorisés en deux flux. Le premier flux a des *slots* et des valeurs prédéfinis, et à chaque tour, le module *DST* essaie de trouver les paires *slots*-valeur les plus appropriées en fonction de l'historique du dialogue. Ce flux traite souvent le *DST* comme un problème de classification multi-class ou multi-hop (M. HENDERSON, THOMSON

et YOUNG 2013, 2014a ; MRKSIC, SÉAGHDHA, THOMSON et al. 2015 ; MRKSIC, SÉAGHDHA, T.-H. WEN et al. 2016 ; ZILKA et JURČÍČEK 2015). Le deuxième flux n'a pas de liste de valeurs de *slots* fixes, donc le module *DST* tente soit d'identifier directement les valeurs pertinentes à partir du contexte du dialogue selon le segment de croyance (*belief span*)(J. CHENG et al. 2020 ; LEI, X. JIN et al. 2018 ; Z. LIN, MADOTTO, WINATA et al. 2020 ; QUAN et XIONG 2020 ; C.-S. WU, MADOTTO et al. 2019), soit de générer ces valeurs en fonction du même contexte (J. CHEN et al. 2020 ; Y. HUANG et al. 2020 ; SHAN et al. 2020 ; Ye WANG, GUO et S. ZHU 2020). L'avantage de cette dernière approche est évidente. Elle réduit non seulement la complexité du modèle et le temps d'inférence du module *DST*, mais facilite également l'entraînement de bout en bout de l'entier système *TOD*. De plus, cette approche est flexible lorsqu'on change de domaine cible (Y. HUANG et al. 2020).

### 2.3.2.3 Module de l'apprentissage des stratégies (Policy Learning - *PL*)

Ce module (*PL*) détermine quelle action le système doit entreprendre à partir de l'état du dialogue fourni par le module *DST*. L'apprentissage supervisé et l'apprentissage par renforcement sont les méthodes principales pour l'apprentissage de la politique de dialogue (Hongshen CHEN et al. 2017). Les politiques apprises de manière supervisée présentent une grande capacité de prise de décision (J. HENDERSON, LEMON et GEORGILA 2008 ; P.-h. SU et al. 2016 ; J. WILLIAMS, ASADI et ZWEIG 2017). Dans certaines tâches spécifiques, le modèle de politique supervisé peut accomplir des tâches avec précision, mais le processus d'apprentissage dépend entièrement de la qualité des données d'entraînement. De plus, les ensembles de données annotés nécessitent un travail humain intensif, et la capacité de décision est limitée par la tâche et le domaine spécifiques, donnant une faible capacité de transfert. Avec la prévalence des méthodes d'apprentissage par renforcement, de plus en plus de systèmes de dialogue axés sur les tâches utilisent l'apprentissage par renforcement pour apprendre la politique (DHINGRA et al. 2016 ; J. HENDERSON, LEMON et GEORGILA 2008 ; B. LIU et LANE 2017 ; J. WILLIAMS, ASADI et ZWEIG 2017). L'apprentissage de la politique de dialogue s'adapte au cadre de l'apprentissage par renforcement, car l'agent de l'apprentissage par renforcement apprend une politique qui associe les états de l'environnement aux actions également. Dans la situation de l'apprentissage par

renforcement des stratégies, un utilisateur humain ou simulateur est nécessaire et l'entraînement dans ce cas est appelé l'apprentissage en ligne (*online learning*). Cependant, cette approche est consommatrice de données, de temps et de travail humain. Pour accélérer le processus d'entraînement et diminuer le coût, J. HENDERSON, LEMON et GEORGILA (2008) utilisent les données d'expert pour restreindre l'exploration initiale de l'espace des actions, tandis que TAKANOBU, R. LIANG et M. HUANG (2020) emploient deux simulateurs pour observer leur interaction et apprendre la stratégie simultanément. Certains travaux proposent d'apprendre les actions et les réponses en même temps puisqu'une réponse hérite la structure de la représentation sémantique donnée par la stratégie (H. LE et al. 2020 ; K. WANG et al. 2020).

#### 2.3.2.4 Module de génération de langue naturelle (Natural Language Generation - NLG)

C'est le dernier module d'une chaîne de traitement de système de dialogue orienté vers la tâche. Il a pour rôle de convertir les actions de dialogue générées par le gestionnaire de dialogue en un énoncé en langage naturel. Les modules traditionnels de *NLG* sont des systèmes de chaîne de traitement. Le processus de génération d'une phrase est divisé par des étapes de la détermination du contenu, de l'ordre et de la structure, puis la planification du syntaxe et la réalisation finale des lexiques et des expressions - tout comme une méthode des perspectives linguistiques classiques (CAHILL et al. 1999 ; REITER 1994). Plus récemment, pour améliorer la performance du module *NLG*, ainsi que pour faciliter la génération et réduire les étapes, des méthodes basées sur l'apprentissage profond sont employées, et le module *NLG* devient un module de bout en bout. En même temps, la génération du langage s'appuie plutôt sur les données que sur les règles linguistiques et les travaux utilisent souvent *CNN*, *RNN* ou *LSTM*, ainsi que leurs variants (DUSEK et JURČÍČEK 2016a,b ; TRAN et NGUYEN 2017 ; T.-H. WEN, GASIC, D. KIM et al. 2015 ; T.-H. WEN, GASIC, MRKSIC, ROJAS-BARAHONA, P.-h. SU, VANDYKE et al. 2016 ; H. ZHOU, M. HUANG et X. ZHU 2016). Les travaux plus récents se sont concentrés sur la fiabilité et la qualité des réponses générées. Certains exploitent la structure d'arbre ou la syntaxe (BAHETI, RITTER et SMALL 2020 ; BALAKRISHNAN et al. 2019), d'autres combinent l'apprentissage supervisé et par renforcement (Y. LI et al. 2020), tandis que plusieurs systèmes

s'appuient sur des modèles de langue pré-entraînés, soit pour s'adapter à de nouvelles tâches (GOLOVANOV et al. 2019), soit pour augmenter les données disponibles (BAHETI, RITTER et SMALL 2020 ; ELDER, O'CONNOR et FOSTER 2020 ; ORABY et al. 2019).

### 2.3.2.5 Méthode de bout en bout (*End-to-end*)

Les modules discutés ci-dessus peuvent obtenir de bonnes performances dans leurs tâches respectives, avec l'aide des progrès récents. Cependant, il existe deux inconvénients importants dans les systèmes modulaires (T. ZHAO et ESKÉNAZI 2016) : (1) Les modules dans de nombreux systèmes en chaîne de traitement ne sont pas toujours différentiables, ce qui signifie que les erreurs finales ne peuvent pas être propagées à chaque module. Dans les systèmes de dialogue réels, le seul signal d'apprentissage provient souvent des interactions utilisateur, les données annotées en états et actions de dialogue demeurant rares et coûteuses à obtenir. (2) Bien que les modules contribuent conjointement au succès d'un système de dialogue, l'amélioration d'un module ne garantit pas nécessairement l'exactitude ou la qualité de la réponse de l'ensemble du système. Cela apporte un entraînement supplémentaire d'autres modules, ce qui est consommateur du temps et du travail humain. De plus, en raison des caractéristiques artisanales des systèmes orientés tâches en chaîne de traitement, tels que les états de dialogue, il est généralement difficile de transférer les systèmes modulaires vers un autre domaine, car les ontologies prédéfinies nécessitent des modifications.

Nous trouvons deux méthodes principales pour faire une architecture *end-to-end* : l'une est d'intégrer chaque module comme composant d'une architecture *end-to-end* et la chaîne de traitement serait un système différentiable de sorte que les paramètres seraient optimisés par la propagation arrière dans un mode *end-to-end* (H. LE et al. 2020) ; l'autre est de prendre un seul module pour à la fois retirer les connaissances et générer les réponses, souvent réalisé par des modèles d'apprentissage multi-tâches.

- **L'architecture en chaîne de traitement *End-to-end*** Bien que de nombreux modules soient facilement différentiables, il reste une tâche qui rend la différenciation difficile : la requête à la base de connaissances. Les méthodes traditionnelles utilisent une requête structurée pour faire correspondre des entrées en fonction de leurs attributs. Le système effectue une analyse

sémantique du message de l'utilisateur pour représenter une requête structurée en fonction de l'objectif de l'utilisateur (X. LI, Y.-N. ( CHEN et al. 2017 ; ROJAS-BARAHONA et al. 2016 ; J. WILLIAMS et ZWEIG 2016)). Comme ce processus n'est pas différentiable, l'architecture entière ne peut pas être entraînée de bout en bout. Pour résoudre ce problème, ERIC et al. (2017) ont utilisé les réseaux de mémoire des paires clé-valeur (MILLER et al. 2016 pour retirer les connaissances reliées. DHINGRA et al. (2016) ont présenté un mécanisme de récupération non discrète (*soft retrieval*) qui repose sur une distribution postérieure continue sur la base de connaissances, remplaçant ainsi les requêtes structurées. Ils ont ensuite intégré ce mécanisme de récupération non discrète dans un cadre d'apprentissage par renforcement pour réaliser un entraînement complet de bout en bout basée sur les réactions des utilisateurs. J. WILLIAMS, ASADI et ZWEIG (2017) ont proposé des réseaux de code hybrides (*HCN*), qui ont encodé les connaissances spécifiques au domaine sous forme de modèles d'action logiciels et systèmes, permettant ainsi la différenciation du module de récupération des connaissances. Ils n'ont pas modélisé explicitement les états du dialogue, mais ont plutôt appris la représentation latente et optimisé le *HCN* en utilisant conjointement l'apprentissage supervisé et l'apprentissage par renforcement. HAM et al. (2020) ont utilisé GPT-2 pour former un chaîne de traitement neuronal et effectuer la prédiction de l'intention, le suivi de l'état du dialogue, l'apprentissage de la politique, la récupération des connaissances et la génération de réponses de manière séquentielle. Le système pouvait facilement interagir avec des systèmes externes car il produisait des résultats intermédiaires explicites de chaque module, ce qui le rendait interprétable. De même, HOSSEINI-ASI et al. (2020) ont construit un chaîne de traitement neuronal avec GPT-2 et ont généré explicitement des résultats pour chaque module neuronal.

- **L'architecture en seul module *End-to-end*** Cette méthode utilise des modèles neuronaux complexes pour représenter implicitement les fonctions clés et intégrer les modules en un seul, en mettant l'accent soit sur les méthodes d'entraînement soit sur l'architecture du modèle, qui sont les clés de la justesse et de la qualité des réponses. W. WANG et al. (2019) ont proposé un cadre d'apprentissage incrémental en construisant

un module d'estimation de l'incertitude pour évaluer la confiance des réponses générées. Si le score de confiance était supérieur à un seuil, la réponse serait acceptée, tandis qu'une réponse humaine serait introduite si le score de confiance était faible. L'agent pouvait également apprendre des réponses humaines en utilisant l'apprentissage en ligne. DAI et al. (2020) ont utilisé le « méta-apprentissage agnostique au modèle » (*MAML*) pour améliorer conjointement l'adaptabilité et la fiabilité avec seulement quelques échantillons d'entraînement dans une tâche de service en ligne de la vie réelle. De manière similaire, QIAN et Z. YU (2019) ont également implémenté le modèle neuronal de bout en bout en utilisant *MAML* pour faciliter l'adaptation de domaine, ce qui permet au modèle de s'entraîner d'abord sur des tâches riches en ressources, puis sur de nouvelles tâches avec des données limitées. Z. LIN, MADOTTO, WINATA et al. (2020) ont proposé une approche de transfert minimal (*minimal transfer learning*) permettant d'adapter des modèles pré-entraînés à grande échelle pour le transfert de domaine dans la tâche de complétion de dialogue. Pour assurer la cohérence séquentielle des réponses générées, J. WU, X. E. WANG et W. Y. WANG (2019) ont entraîné un module non supervisé de détection des incohérences dans l'ordre des réponses. Ce module détectait si une paire d'énoncés était ordonnée ou non pour guider l'agent de complétion de tâche vers la génération de réponses plus cohérentes. BALAKRISHNAN et al. (2019) ont introduit une méthode de décodage contrainte pour améliorer la qualité sémantique des réponses générées par le système de bout en bout proposé. De nombreux systèmes orientés tâches de bout en bout ont utilisé un module de mémoire pour stocker les faits de connaissances pertinents et l'historique du dialogue. X. CHEN, J. XU et B. XU (2019) ont fait valoir qu'un seul module de mémoire n'était pas suffisant pour une récupération précise. Ils ont utilisé deux modules de mémoire à long terme pour stocker les paires de connaissances et l'historique du dialogue respectivement, puis une mémoire de travail a été appliquée pour contrôler la génération de tokens. Yichi ZHANG et al. (2020) ont proposé le modèle d'état de croyance latente (*Latent Belief State*), qui traitait les états de dialogue comme des variables latentes discrètes pour réduire la dépendance à l'égard des étiquettes *DST* au niveau du tour. Pour résoudre le problème



de l'insuffisance des données dans certaines tâches, S. GAO et al. (2020) ont augmenté le modèle de génération de réponses avec un modèle de paraphrase dans leur système de bout en bout. Le modèle de paraphrase a été entraîné conjointement avec l'ensemble du cadre et visait à augmenter les échantillons d'entraînement. S. YANG, R. ZHANG et ERFANI (2020) ont exploité les informations de structure de graphe à la fois d'un graphe de connaissances et de l'arbre de dépendance du contexte de dialogue. Ils ont proposé une architecture de réseaux de neurones récurrentes (RNN) pour apprendre des représentations sur le graphe et ont effectué un raisonnement multi-sauts pour exploiter les liens d'entités dans le graphe de connaissances.

## 2.4 Système de recommandation conversationnel

Parmi les tâches d'un agent conversationnel *TOD*, il y a une tâche spécifique, qui est la tâche principale dans notre scénario : la recommandation des produits. Ainsi, nous étudions aussi le sujet des systèmes de recommandation conversationnel (Conversational Recommender System - *CRS*). Un *CRS*, par sa définition, est un système d'interactions qui s'adapte aux préférences dynamiques des utilisateurs et agit en fonction de leurs besoins actuels grâce à des interactions en temps réel à plusieurs tours (C. GAO et al. 2021).

Les systèmes de recommandation traditionnels, c'est-à-dire les modèles de recommandation statiques et hors ligne, prédisent principalement la préférence d'un utilisateur pour un article en analysant ses comportements passés, tels que l'historique des clics, des visites et des commentaires sur des articles. Ces méthodes ne s'adaptent pas en temps réel aux interactions de l'utilisateur, mais sont entraînées et évaluées sur des données historiques. Les approches initiales comprennent le filtrage collaboratif, les machines de factorisation et les arbres de décision. Récemment, des modèles plus avancés ont été développés, incluant le filtrage collaboratif neuronal (Xiangnan HE, L. LIAO et al. 2017), le réseau d'intérêt profond (G. ZHOU et al. 2017), le modèle profond basé sur les arbres (H. ZHU et al. 2018) et les réseaux de convolution graphique (Graph Convolutional Networks, GCN) (Xiangnan HE, DENG et al. 2020 ; F. WU et al. 2019 ; YING et al. 2018).

Contrairement aux méthodes traditionnelles statiques des systèmes de re-

commandation, les CRS interagissent pour solliciter la préférence actuelle d'un utilisateur et modifient les recommandations en fonction des retours instantanés de l'utilisateur. Depuis la naissance des systèmes de recommandation, les chercheurs ont réalisé l'importance de l'interaction homme-machine. Certaines études proposent des systèmes de recommandation interactifs (Haokun CHEN et al. 2019 ; C. HE, PARRA et VERBERT 2016 ; Huazheng WANG, Q. WU et Hongning WANG 2017 ; S. ZHOU et al. 2020) et des systèmes de recommandation basés sur les critiques (BURKE, HAMMOND et YOUNG 1997 ; L. CHEN et PU 2012 ; K. LUO, SANNER et al. 2020 ; K. LUO, H. YANG et al. 2020 ; PU et FALTINGS 2004 ; SMYTH et MCGINTY 2003 ; TOU et al. 1982 ; TVERSKY et SIMONSON 1993), qui peuvent être considérés comme des formes précoces de CRS, car ils se concentrent sur l'amélioration de la stratégie de recommandation en ligne en exploitant les commentaires en temps réel des utilisateurs sur les articles recommandés précédemment. Les systèmes de recommandation interactifs donnent les recommandations selon la réaction de l'utilisateur indiquant, à chaque tour, si la personne apprécie la recommandation et dans quelle mesure. Mais ces systèmes souffrent souvent d'une faible efficacité à cause de la grande quantité des articles. Les systèmes basés sur les critiques règlent ce problème en demandant à l'utilisateur d'évaluer les attributs liés aux articles au lieu des articles spécifiques. Comme une version améliorée de ces systèmes, les CRSs profitent de la nature des conversations multi-tours. Ils peuvent décider à quel moment recommander ou continuer les questions, ou inférer la préférence de l'utilisateur depuis son intention pour réduire rapidement la liste des candidats recommandés. Certes, il y existe des CRSs utilisant l'architecture des agents conversationnels pour traiter les conversations, mais il faut indiquer que dans les CRSs, les interactions peuvent être de toutes formes (e.g. buttons, champs de formulaire, langue naturelle).

Dans la figure 2.6, nous résumons quelques défis principaux pour un CRS. D'abord, comment obtenir la préférence d'utilisateur en posant de bonnes questions ? Deux problématiques importantes sont abordées : (1) Quoi demander ? (2) Comment ajuster les recommandations en fonction de la réponse de l'utilisateur ? La première question se concentre sur la construction de questions pour obtenir autant d'informations que possible ; la seconde exploite les informations contenues dans la réponse de l'utilisateur pour formuler des recommandations plus

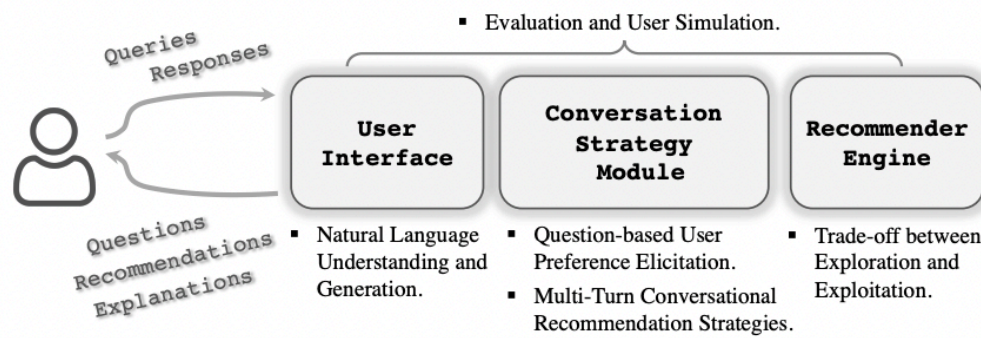


FIG. 2.6 : La structure d'un système de recommandation conversationnel, CRS (C. GAO et al. 2021).

appropriées. Pour résoudre ces deux problématiques, certains chercheurs choisissent de poser des questions sur les articles eux-mêmes (CHRISTAKOPOULOU, RADLINSKI et HOFMANN 2016 ; LOEPP, HUSSEIN et ZIEGLER 2014 ; MANGILI et al. 2020 ; VENDROV et al. 2020 ; Tong YU, SHEN et H. JIN 2019 ; X. ZHAO, W. ZHANG et J. WANG 2013 ; L. ZOU et al. 2020) tandis que d'autres préfèrent poser des questions sur les attributs des articles (CHRISTAKOPOULOU, BEUTEL et al. 2018 ; LEI, Xiangnan HE et al. 2020 ; LEI, G. ZHANG et al. 2020 ; MANGILI et al. 2020 ; Y. SUN et Yi ZHANG 2018 ; G. WU et al. 2019 ; H. YANG et al. 2021 ; Yongfeng ZHANG et al. 2018 ; K. ZHOU, W. X. ZHAO et al. 2020). Deuxièmement, étant donnée des conversations aux tours multiples, comment trouver de bonnes stratégies dynamiquement ? Une problématique importante concerne le moment où poser des questions et le moment où faire des recommandations, c'est-à-dire laisser le modèle choisir entre (1) continuer à poser des questions afin de réduire davantage l'incertitude de préférence, et (2) générer une recommandation basée sur l'estimation de la préférence actuelle de l'utilisateur. En général, le système devrait viser à faire une recommandation réussie en utilisant le moins de tours possible, car les utilisateurs perdent patience après trop de tours (LEI, Xiangnan HE et al. 2020). De plus, certaines stratégies conversationnelles sophistiquées tentent de diriger activement les dialogues (BALARAMAN et MAGNINI 2020 ; W. WU et al. 2019), ce qui peut introduire divers sujets et tâches dans les CRS (M. LEWIS et al. 2017 ; X. WANG et al. 2019 ; K. ZHOU, Y. ZHOU et al. 2020). Certaines recherches (LEI, Xiangnan HE et al. 2020 ; Y. SUN et Yi ZHANG 2018 ; K. XU et al. 2021 ; Yongfeng ZHANG et al. 2018) développent des stratégies

explicites tandis que certaine d’autre essayent de mener les conversations aux divers sujets (Z. LIU et al. 2020 ; K. ZHOU, Y. ZHOU et al. 2020). Le troisième défi porte sur la compréhension et la génération de langue tout comme dans un agent conversationnel. La plupart des travaux se concentrent sur le développement d’un agent conversationnel en architecture *end-to-end* pour traiter les entrées sous forme de langue naturelle (Q. CHEN et al. 2019 ; Raymond LI et al. 2018 ; MOON et al. 2019b ; H. XU et al. 2020 ; K. ZHOU, W. X. ZHAO et al. 2020). Puisque tous les systèmes de recommandation sont confrontés au dilemme entre exploration et exploitation, les CRS ne font pas exception, car chaque utilisateur peut interagir uniquement avec quelques articles parmi l’ensemble complet des données. Un grand nombre d’articles susceptibles d’intéresser un utilisateur resteront invisibles pour lui. Pour les utilisateurs en démarrage à froid (qui viennent de rejoindre le système et ont zéro ou très peu d’interactions), le problème est particulièrement grave. Grâce à la nature interactive, les SRI peuvent explorer activement les articles invisibles pour mieux capturer la préférence de l’utilisateur. De cette manière, les utilisateurs peuvent exprimer leurs intentions et obtenir des recommandations mieux personnalisées. Cependant, le processus d’exploration a un coût. Comme les utilisateurs n’ont qu’un temps et une énergie limités pour interagir avec le système, une exploration ratée gaspillera du temps et perdra l’opportunité de faire des recommandations précises. De plus, exposer des articles non pertinents nuit à la préférence de l’utilisateur, par rapport à l’exploitation de la préférence déjà capturée en recommandant les articles ayant une forte confiance (GILOTTE et al. 2018 ; L. LI, J. Y. KIM et ZITOUNI 2015 ; SCHNABEL et al. 2018). Une approche largement étudiée repose sur les « *multi-armed bandits* », couramment utilisés dans les systèmes de recommandation pour modéliser le compromis entre exploration et exploitation, et ayant inspiré de nombreux travaux récents (CHRISTAKOPOULOU, RADLINSKI et HOFMANN 2016 ; Shijun LI et al. 2021 ; Tong YU, SHEN et H. JIN 2019 ; X. ZHANG et al. 2020) l’appliquent dans CRS. Les performances d’un CRS sont souvent évaluées par les tests A/B (GILOTTE et al. 2018) ou les simulateurs d’utilisateur (J. HUANG et al. 2020 ; W. SUN et al. 2021 ; Shuo ZHANG et BALOG 2020).

## 2.5 Personnalisation

Dans le cadre des systèmes de dialogue, la notion de *personnalité* se réfère à l'ensemble des traits ou caractéristiques stables qui définissent la manière dont un agent ou un utilisateur interagit et s'exprime. La personnalité de l'utilisateur influence la manière dont le système comprend et répond à ses requêtes, tandis qu'une personnalité cohérente du système est essentielle pour garantir la crédibilité et la continuité des échanges. Cela devient un des plus grands sujets surtout pour les systèmes *ODD*, car les réponses sont entièrement apprises à partir des données d'entraînement. Par exemple, si le système ne maintient pas de cohérence dans les informations personnelles qu'il fournit (âge, nom, préférences, etc.), il peut produire des réponses contradictoires lorsque l'utilisateur pose plusieurs fois la même question, ce qui nuit à l'expérience utilisateur. Ainsi, pour un agent conversationnel basé sur les données, il est nécessaire de prendre conscience de son rôle et de répondre en fonction d'une personnalité fixe.

Modéliser explicitement la personnalité est la principale stratégie dans les travaux récents. Q. LIU et al. (2020) ont proposé un générateur de dialogue basé sur la personnalité composé d'un Récepteur (*Receiver*) et d'un Émetteur (*Transmitter*). Le récepteur était responsable de modéliser la personnalité de l'interlocuteur à travers plusieurs échanges de conversation, tandis que l'émetteur générait des énoncés en se basant sur la personnalité de l'agent et de l'interlocuteur, ainsi que sur le contenu de la conversation. H. KIM, B. KIM et G. KIM (2020) proposent un générateur de dialogue qui incorpore la personnalité sans supervision explicite. Un *auditeur imaginaire* estime la personnalité exprimée à partir de la séquence générée, puis un *locuteur conscient de soi* produit le texte final en alignant la génération sur cette personnalité. Pour contrôler la personnalité dans les réponses générées par un grand modèle de langue (*LLM*), il est nécessaire d'utiliser des *prompts*, c'est-à-dire des instructions ou des descriptions intégrées à l'entrée du modèle pour guider la génération (voir la section 2.3.1.4). Par exemple, BOYD et al. (2020) ont utilisé un GPT-2 augmenté pour raisonner sur les conversations passées et modéliser la personnalité de l'acteur cible. Pour obtenir un agent ayant une personnalité spécifique, il faut conditionner le modèle sur des descriptions explicites telles que « Je suis une personne généreuse ». Néanmoins, cette approche implique une modélisation manuelle des traits de

personnalité, ce qui requiert un travail manuel considérable. Z. LIN, MADOTTO, C.-S. WU et al. (2019) ont proposé d'utiliser le « méta-apprentissage agnostique au modèle » (*MAML*) pour s'adapter à de nouvelles personnalités avec seulement quelques exemples d'entraînement et sans avoir besoin de descriptions de personnalité. MAJUMDER, JHAMTANI et al. (2020) se sont appuyés sur des sources de connaissances externes pour étendre les descriptions de personnalité actuelles, de sorte à obtenir des descriptions de personnalité plus riches et à associer ces descriptions actuelles à des faits de bon sens. H. SONG, Yan WANG, W. ZHANG, X. LIU et al. (2020) ont argumenté que les systèmes traditionnels étaient des systèmes à un seul stade et que les réponses qu'ils généraient contenaient encore de nombreux mots incohérents avec la personnalité. Pour résoudre ce problème, ils ont proposé une architecture en trois étapes pour assurer la cohérence de la personnalité : un mécanisme de « génération-suppression-réécriture » a été mis en œuvre pour supprimer les mots inacceptables générés dans les réponses prototypes et les réécrire de manière cohérente avec la personnalité.

## 2.6 Dialogues multimodaux

Le dialogue multimodal, surtout visuel, peut être considéré comme un processus de raisonnement en plusieurs étapes sur une série de questions. GAN, Y. CHENG et al. (2019) ont proposé un modèle qui apprend une représentation sémantique de la question en combinant à la fois l'historique du dialogue et l'image associée. Cette représentation est ensuite mise à jour de manière récurrente à chaque étape du dialogue, permettant au modèle de capturer l'évolution contextuelle et visuelle de la conversation. SHUSTER, JU et al. (2019) ont introduit un benchmark pour les dialogues multimodaux, comprenant plusieurs tâches centrées sur l'image. Tao YU et JOTY (2020) ont utilisé un R-CNN comme encodeur d'image et ont fusionné les informations visuelles et de dialogue à l'aide d'un modèle BERT spécialisé pour le traitement conjoint des modalités visuelles et textuelles (Visual-Dialogue BERT, ou VD-BERT). L'architecture proposée a permis des interactions suffisantes entre le dialogue à plusieurs tours et les images.

Certains chercheurs ont des opinions différentes sur l'efficacité de l'historique du dialogue dans le dialogue visuel. TAKMAZ et al. (2020) ont fait l'hypothèse que de nombreuses expressions étaient déjà mentionnées dans les tours précédents, et ils ont construit un modèle de dialogue visuel basé à la fois sur l'image et l'historique

de conversation. Ils ont ensuite prouvé que de meilleures performances étaient obtenues lorsque le modèle était ancré sur le contexte du dialogue. Cependant, AGARWAL et al. (2020) ont soutenu que, bien que le modèle de dialogue visuel avec historique du dialogue puisse obtenir de meilleurs résultats, en réalité seule une petite proportion de cas bénéficie de l'historique. De plus, ils ont prouvé que les métriques d'évaluation existantes pour le dialogue visuel favorisaient les réponses génériques.

Certains chercheurs ont également porté leur attention sur le pré-entraînement visuel-linguistique, inspirés par les premiers travaux de pré-entraînement des modèles. Pour cela, il existe principalement deux types de conceptions de modèles selon leur architecture et leur stratégie de fusion des modalités :

1. **Le modèle à flux unique** (*single-stream model*) : toutes les modalités d'entrée (par exemple le texte et l'image) sont fusionnées dès l'encodage et traitées dans un flux unique (ALBERTI et al. 2019 ; Y.-C. CHEN et al. 2019 ; GAN, Y.-C. CHEN et al. 2020 ; G. LI et al. 2019 ; L. H. LI et al. 2019 ; X. LI, YIN et al. 2020 ; W. SU et al. 2019 ; L. ZHOU et al. 2019).
2. **Le modèle à deux flux** (*dual-stream model*) : chaque modalité est encodée séparément dans deux flux parallèles, puis les représentations sont combinées à un stade ultérieur (J. LU, D. BATRA et al. 2019 ; J. LU, GOSWAMI et al. 2019 ; TAN et BANSAL 2019 ; F. YU et al. 2020).

Pour le modèle à flux unique, G. LI et al. (2019) ont utilisé un modèle BERT pour traiter la concaténation d'objets et de mots, et l'ont pré-entraîné avec des tâches générales. Des méthodes similaires ont été proposées par Y.-C. CHEN et al. (2019) et QI et al. (2020), mais avec plus de tâches de pré-entraînement et de plus grands ensembles de données. Avec une technique d'entraînement adversaire, GAN, Y.-C. CHEN et al. (2020) ont amélioré ce modèle. W. SU et al. (2019) ont utilisé la même architecture, mais ont incorporé des données à modalité unique et ont pré-entraîné le détecteur d'objets. Au lieu d'utiliser des objets reconnus, Z. HUANG et al. (2020) ont cherché à utiliser directement les pixels. Les étiquettes d'objets ont été utilisées par X. LI, YIN et al. (2020) pour améliorer l'alignement entre modalités croisées. L. ZHOU et al. (2019) ont proposé un modèle à flux unique qui apprend à la fois la génération de légendes et les tâches de VQA (questions-réponses visuelles) en même temps.

Pour le modèle à deux flux, TAN et BANSAL (2019) ont proposé un modèle à

deux flux avec co-attention et ont utilisé uniquement des données du domaine pour entraîner le modèle. J. LU, D. BATRA et al. (2019) ont introduit une architecture similaire avec un modèle de co-attention plus complexe, nommé ViLBERT (*Vision-Language BERT*), qu'ils ont pré-entraîné sur des données hors domaine. Par la suite, J. LU, GOSWAMI et al. (2019) ont amélioré ViLBERT en utilisant un apprentissage multi-tâches. F. YU et al. (2020) ont récemment ajouté un graphe de scène (*scene graph*) au modèle, ce qui a amélioré les performances. Un *scene graph* est une représentation structurée des objets présents dans une image ainsi que des relations entre eux. En dehors de ces études, A. SINGH, GOSWAMI et PARIKH (2020) ont étudié l'impact de la sélection de l'ensemble de données de pré-entraînement sur les performances des tâches ultérieures.

L'annotation du dialogue visuel est laborieuse, et donc les ensembles de données sont rares. Récemment, certains chercheurs ont essayé de résoudre le problème du manque de données. SHUSTER, HUMEAU et al. (2018) ont collecté un ensemble de données de conversations humain-humain ancrées dans l'image, dans lesquelles les interlocuteurs sont invités à jouer un rôle basé sur une humeur émotionnelle ou un style offert, car l'utilisation de telles caractéristiques est également un facteur important pour l'engagement. COGSWELL et al. (2020) ont proposé une méthode pour utiliser les données VQA lors de l'adaptation à une nouvelle tâche, minimisant ainsi le besoin de données de dialogue qui sont coûteuses à annoter.



Deuxième partie

Méthode



## Chapitre 3

---

### Méthode proposée

#### 3.1 Méthode proposée

L'objectif principal de nos travaux est de développer un système de dialogue capable de répondre aux questions des utilisateurs en fonction de leur profil et de recommander des produits, tout en étant capable de recevoir et de répondre avec des images. Pour réaliser un tel système de dialogue, plusieurs éléments essentiels sont nécessaires :

- **Collecte et génération de données** Nous avons besoin de suffisamment de données, y compris des données sur les produits, un lexique de termes, les profils des utilisateurs et des dialogues. Les dialogues sont particulièrement importants, donc nous devons collecter ou générer une quantité suffisante de données de dialogues annotés pour l'entraînement. Nous allons proposer différentes méthodes pour collecter, générer et annoter les données de dialogues telles que :
  - les dialogues entre humains
  - les simulateurs d'utilisateurs et de systèmes
  - les interactions humain-machine

Nous allons évaluer la qualité et le coût des données obtenues à l'aide de méthodes d'évaluation appropriées.

- **Entraînement du système de dialogue** Le système doit être capable de comprendre et de répondre aux questions des utilisateurs, ainsi que de trouver des produits correspondants dans la base de données de produits. En plus des capacités de base d'un système de dialogue, notre système devrait être capable de traiter des informations multimodales, être personnalisé en fonction des caractéristiques des utilisateurs et être précis dans ses recommandations et réponses en utilisant des connaissances externes. Nous

comparerons différentes architectures pour construire ce système, comme l'utilisation de modules traditionnels ou une approche de bout en bout, introduits dans le chapitre 2.3.2. Nous étudierons également la question de savoir si les informations visuelles, les connaissances externes et les caractéristiques personnelles des utilisateurs devraient être intégrées dans des modules séparés (*NLU*, *DST*, *Policy Learning*, *NLG*), ou si elles seraient traitées dans les modules indépendants en parallèle des autres modules basiques, ou encore si elles devraient être fournies en tant qu'entrée avec l'entrée des dialogues à un modèle de bout en bout. Nous évaluerons également ces méthodes en termes de précision, de coût et de stabilité. Enfin, nous explorons un mécanisme original d'auto-alignement itératif, combinant deux simulateurs : un simulateur d'utilisateur, qui modélise les préférences et les comportements d'achat, et un simulateur de vendeur, qui génère des descriptions de produits à partir de la perspective utilisateur. En confrontant ces deux pôles, le système apprend à générer un langage persuasif mais équitable, et à ajuster ses prédictions en fonction de l'utilité perçue par l'utilisateur simulé.

- **Évaluation et analyse** Nous examinerons également les aspects de l'expérience utilisateur et de la facilité de développement pour discuter de l'efficacité et des nécessités de notre système. L'objectif ultime est de développer un système de dialogue convivial et précis, qui soit adapté aux besoins spécifiques des utilisateurs et qui offre une expérience utilisateur satisfaisante tout en étant facile à développer et à maintenir. Par analyser les performances de différentes d'expérimentation, nous répondrons aux problématiques de faisabilité et de nécessité proposées dans le dernier chapitre : faisabilité de développer et de déployer un agent conversationnel complexe ainsi que nécessité sous différents aspects, de la personnalisation, de la multimodalité ou même d'un système de dialogue par rapport à un système de récupération. Dans cette optique, nous avons élaboré notre plan d'expérimentation, découpé en plusieurs étapes, avec l'accent mis sur les dimensions clés telles que l'information, la modalité, le modèle et la forme d'interactivité. Ces dimensions évoluent chacune graduellement de la simplicité à la complexité. L'information englobe les éléments tels que le titre du produit, les images du produit, les caractéristiques du produit, la

description du produit, les informations sur les stocks, les caractéristiques des utilisateurs, les descriptions des utilisateurs, l'historique des achats des utilisateurs, etc. Les modalités comprennent à la fois la composante visuelle et textuelle. Les modèles englobent à la fois des modèles traitant uniquement le texte et des modèles multimodaux, et nous examinons leurs méthodes d'entraînement, y compris l'apprentissage, l'affinage, le réglage des prompts et l'exploitation d'outils externes. En ce qui concerne la forme, nous allons de simples recherches, classements et appels API vers des systèmes de dialogue. Notre plan d'expérimentation consiste à progresser graduellement sur ces dimensions, afin de comparer les résultats finaux et de répondre à nos questions sur la faisabilité et les nécessités posées par notre problématique.

### 3.2 Plan d'expérimentation

Nous rappelons qu'un système de dialogue pourrait être décrit et implémenté à travers une fonction de correspondance appelée  $\varphi$ . Cette fonction  $\varphi$  guidera l'ensemble de nos démarches théoriques et expérimentales tout au long de nos travaux. Pour commencer, un système de dialogue des plus simples peut être décrit comme  $R = \varphi(U)$  où  $U$  est simplement le message de l'utilisateur et  $R$  est la réponse de l'agent. Dans ce cas-là, le système n'est qu'un agent question-réponse à un seul tour. Sur cette base, des éléments tels que l'historique des conversations  $H$ , une base de connaissances  $K$ , les personnalités ou préférences des parties  $P$ , entre autres, sont continuellement intégrés. Ceci vise à rendre les réponses du robot  $R$  plus précises. Par ailleurs, des efforts sont déployés pour augmenter la modalité de l'information, tels que les informations audiovisuelles, les informations structurées, afin d'enrichir et d'élargir la portée de l'information dans la conversation. Ainsi, notre plan repose sur la progression graduelle à travers plusieurs dimensions clés, ce qui nous permettra d'évaluer l'impact de chaque élément sur l'ensemble du système.

Tout d'abord, nous examinons la dimension de l'information. Il s'agit de la base de notre expérimentation, car nous utilisons une variété de données à part de l'historique des conversations  $H$ , notamment une base de connaissances  $K$ , qui comprend les titres de produits, les images, les caractéristiques, les descriptions, les niveaux de stock, etc. ; à la fois les personnalités ou préférences

des parties  $P$ , par exemple, les personnalités de l'utilisateur et de l'agent, les descriptions d'utilisateurs et son historique des achats. Cette dimension nous permet de déterminer quelles informations sont essentielles pour l'agent et dans quelle mesure elles contribuent à l'efficacité du système.

Ensuite, nous abordons la dimension de la modalité, qui combine à la fois la composante visuelle et textuelle. Il est important de comprendre comment l'information visuelle, telle que les images de produits, influence les recommandations et les interactions avec les utilisateurs par rapport au texte seul.

La dimension du modèle est également cruciale. Nous examinons à la fois des modèles traitant uniquement le texte et des modèles multimodaux. De plus, nous explorons différentes méthodes d'entraînement, notamment l'apprentissage, l'affinage, le réglage des prompts et la planification d'outils. Cela nous permettra de déterminer quelles approches sont les plus efficaces pour notre contexte.

Enfin, nous arrivons à la dimension de la forme, qui représente les différentes étapes de l'interaction utilisateur. Nous passons des méthodes simples telles que la recherche et le classement à des systèmes de dialogue complets. Cela nous permet de voir comment la complexité de l'interaction affecte l'expérience utilisateur et l'efficacité du système.

Pour entraîner notre modèle, nous avons proposé un nouveau cadre de génération-simulation pour aligner les recommandations de produits sur les principes de prise de décision rationnelle. Cette architecture d'auto-alignement nous permet d'optimiser notre modèle sur l'objectif de l'utilité au lieu de la seule vraisemblance linguistique. Elle repose sur deux simulateurs entraînés à partir du même LLM :

- un simulateur d'utilisateur, qui évalue si une description de produit correspond à ses attentes et décide, de manière explicite, d'acheter ou non le produit ;
- un simulateur de vendeur (ou générateur de contenu), qui produit les descriptions des produits.

Le simulateur d'utilisateur joue ainsi le rôle d'un superviseur en fournissant un retour direct au générateur de contenu, ce qui permet d'améliorer progressivement la qualité et la pertinence des recommandations.

En superposant ces dimensions graduellement, nous serons en mesure de comparer les résultats finaux à chaque étape. Cela nous aidera à répondre aux questions essentielles de notre problématique, à savoir la faisabilité de la person-

nalisation, la nécessité de la multimodalité et le rôle des agents conversationnels par rapport à d'autres formes d'interaction. Notre plan d'expérimentation est conçu pour apporter des réponses précises et éclairées à ces questions, dont nous espérons qu'elles contribueront à l'avancement de notre compréhension des systèmes de dialogue pour le commerce électronique de mode.





## Chapitre 4

---

### Génération de données

Cette partie vise à explorer en détail le processus de création d'un ensemble de données pour notre système de dialogue, avec un accent particulier sur la construction de la base de connaissances et la génération de données de dialogue.

Nous débutons par l'examen des différentes ressources numériques dans le domaine de la mode dans la section 4.1, incluant les sites de commerce électronique, les contenus informatifs et les ensembles de données existants. Nous expliquons comment nous avons collecté ces données de manière systématique.

Ensuite, nous présentons la construction d'une base de connaissances structurée, décrite en détail dans la section 4.2. Ce corpus inclut des informations sur les produits, une terminologie et une taxonomie spécifiques au domaine ainsi que des connaissances contextuelles telles que des techniques de style, des aspects esthétiques et des informations culturelles et commerciales. Nous abordons également l'intégration des personnalités des utilisateurs pour enrichir ce corpus.

Enfin, nous passons à la génération de dialogues dans la section 4.3. Nous analysons les approches courantes pour obtenir des dialogues, en comparant la collecte de dialogues existants à la génération de nouveaux dialogues. Nous décrivons nos méthodes pour générer des dialogues en utilisant à la fois des automates et des grands modèles de langage (*LLMs*), et incluons des exemples générés concrets.

#### 4.1 Obtenir des données originales

Dans cette section, nous détaillons le processus d'acquisition de données originales nécessaires à la construction de notre base de données. Nous commençons par examiner les différentes ressources numériques disponibles dans le domaine de la mode, y compris les sites de commerce électronique, les contenus informa-

tifs et les ensembles de données existants. Ensuite, nous décrivons la méthode de la collecte des données, montrant les outils que nous avons utilisés pour extraire, nettoyer et organiser ces données.

#### 4.1.1 *Ressources numériques dans le domaine de la mode*

Nous définissons le domaine de la mode en incluant les vêtements, la bijouterie, la joaillerie, les montres, ainsi que d'autres éléments tels que les accessoires, la maroquinerie, la lingerie, les parfums et les cosmétiques. Ce domaine combine des aspects commerciaux, esthétiques et historiques. On trouve de nombreuses formes de ressources numériques dans le domaine de la mode, certaines concernent les produits et d'autres se concentrent sur les aspects humains. En raison du caractère à la fois visuel et commercial du domaine de la mode, les attributs des produits (tels que la couleur, la taille, la matière, le style ou la marque) et leurs images associées constituent naturellement une forme d'annotation implicite, offrant ainsi une source de données précieuse pour l'entraînement des modèles. Du côté humain, les ressources se manifestent sous la forme de textes subjectifs ou expressifs : les descriptions narratives dans les magazines, les dialogues entre vendeurs et clients, les critiques d'experts ou encore les commentaires des consommateurs. Ces données traduisent la dimension culturelle, émotionnelle et sociale de la mode, et complètent les informations plus objectives issues des produits eux-mêmes.

Nous divisons grossièrement les ressources numériques en deux catégories : les données structurées et les données non structurées. Les données structurées comprennent les connaissances de l'industrie, telles que la terminologie ou la taxonomie de divers types de produits ; les descriptions au format CSV de certains produits, telles que les catalogues ou les enregistrements de stocks, entre autres. Les données non structurées comprennent les images et le langage, notamment les dialogues entre l'acheteur et le vendeur, les entretiens avec des experts de la mode, les commentaires de produits, les titres de produits et les descriptions longues des produits, entre autres.

Où trouver ces données ? Des pages de commerce électronique comportant des informations et des images des produits, des revues et des magazines décrivant la conception, l'artisanat, et même la philosophie des marques, etc. D'autres sources de données pourraient être les publicités ou les réseaux sociaux.

Outre ces données brutes, de nombreux ensembles de données et travaux de recherche publiés dans le milieu académique sont également des ressources précieuses. La plupart des jeux de données dans le domaine de la mode sont de forme multimodale. Ils visent soit à capturer la sémantique de l'image, comme *Farfetch* (MARCELINO et al. 2018), *FashionAI* (X. ZOU et al. 2019) et *Persuaide !* (MUNIGALA, TAMILSELVAM et SANKARAN 2017), soit à analyser les commentaires des produits ou des images (NARAYANAN, SHI et RUBINSTEIN 2011). Par rapport à l'agent conversationnel, ces tâches sont plutôt davantage axées sur la compréhension visuelle. Par exemple, *FashionAI* a développé un système d'attributs hiérarchiques à grain fin pour mieux décrire les produits et leurs images, tandis que *Persuaide !* génère des textes persuasifs pour rendre le produit plus attirant pour la clientèle.

Certains travaux consistent à chercher des objets à partir de textes ou de conversations (CROOK et al. 2019 ; KOTTUR, MOON et al. 2021 ; A. SAHA, KHAPRA et SANKARANARAYANAN 2018 ; H. WU et al. 2021a ; YUAN et LAM 2021). Ils s'appuient souvent sur la comparaison des produits candidats au niveau de leurs attributs et la proposition d'autres alternatives selon les attributs préférés. Par exemple, dans les dialogues de *FashionIQ* (H. WU et al. 2021b), l'utilisateur fournit des commentaires en langage naturel et l'agent retourne des images récupérées. Ce travail vise à offrir des expériences plus naturelles, expressives et conviviales lors de la recherche d'articles que les interfaces classiques basées sur des mots-clés. Le jeu de données fournit également un grand nombre de paires d'images et de leurs légendes associées.

En un mot, le domaine de la mode est assez bien doté en ressources numériques. Cependant, ces données doivent être reformatées et filtrées afin de mieux correspondre à nos scénarios et à d'autres tâches de TAL.

##### 4.1.2 Collecte des données

Nous groupons les principales sources de données dans le domaine de la mode : les pages de commerce électronique, le contenu informatif (articles, vidéos, etc.), les œuvres littéraires et cinématographiques, les médias sociaux et les ensembles de données publics. En général, les ensembles de données publics sont l'endroit où nous pouvons trouver rapidement des données de haute qualité. Nous allons détailler comment nous collectons et formalisons les données selon ces différentes

sources.

#### 4.1.2.1 Site de commerce électronique

Les sites de commerce électronique affichent des informations produit riches et variées. Nous utilisons principalement des scripts Python pour extraire ces informations des pages web. Le travail de collecte repose sur le fait que chaque site a une structure différente, ce qui implique que le type et la quantité d'informations disponibles varient. Nous commençons donc par diviser les sites de commerce électronique en deux catégories principales : les plateformes et les sites indépendants. Les plateformes se présentent sous deux types différents : celles qui sont les sites officiels de grands magasins physiques ou virtuels en soi (comme *Lafayette*<sup>1</sup>, *Le Bon Marché*<sup>2</sup>, *Farfetch*<sup>3</sup> et *Shein*<sup>4</sup>) et celles qui servent d'intermédiaires pour les transactions entre vendeurs et acheteurs (comme *Amazon*<sup>5</sup>, *Etsy*<sup>6</sup>, *Zalando*<sup>7</sup>, etc).

Outre les sites de grands magasins et de plateformes, de nombreuses marques de mode choisissent de créer leurs propres sites indépendants. Afin de refléter leur style unique, certaines marques développent des sites web avec des effets visuels complexes, ce qui complique la collecte de données. De plus, pour des raisons de marketing, ces marques mettent souvent l'accent sur les aspects visuels et proposent généralement des descriptions de produits concises. En outre, certaines marques utilisent des expressions spéciales ou littéraires pour décrire leurs produits, rendant ces descriptions uniques mais posant des défis supplémentaires pour la compréhension et la catégorisation automatique des produits par les machines.

Une autre catégorie de sites indépendants concerne les marques émergentes ou les commerçants en ligne transfrontaliers. Ces marques préfèrent souvent utiliser des formats standardisés et concis pour construire leurs sites web, tels que le format *Shopify*<sup>8</sup>. Ces types de sites web sont particulièrement utiles pour notre

---

<sup>1</sup><https://www.gallerieslafayette.com/>

<sup>2</sup><https://www.24s.com/>

<sup>3</sup><https://www.farfetch.com/>

<sup>4</sup><https://www.shein.com/>

<sup>5</sup><https://www.amazon.com/>

<sup>6</sup><https://www.etsy.com/>

<sup>7</sup><https://www.zalando.com/>

<sup>8</sup><https://www.shopify.com/>

recherche. En effet, une fois que l'utilisation de *Shopify* est identifiée, la collecte et la compréhension du contenu des pages produits deviennent beaucoup plus simples et efficaces.

Contrairement à la plupart des sites de commerce électronique, *Shein* dispose de sa propre API ouverte au public, permettant aux développeurs de récupérer les données produits à l'aide de leur *skuCode* (un type d'ID). Cette méthode nous évite l'effort d'analyser des pages web et renvoie directement les informations sur les produits au format JSON.

Pour les autres sites de commerce électronique, nous utilisons la méthode de web scraping, en employant des outils et bibliothèques Python tels que *BeautifulSoup* et *Scrapy*. Prenons *24s* comme exemple, pour extraire des données produits, voici les étapes générales que nous suivons :

1. Identification des URL cibles : Nous déterminons les URL des pages de catégories sur *Shein* à partir desquelles nous souhaitons extraire les URL des produits.
2. Envoi de requêtes HTTP : Nous utilisons la bibliothèque comme *Requests* pour envoyer des requêtes HTTP à ces URL, simulant ainsi la navigation d'un utilisateur sur le site.
3. Analyse du code HTML : Une fois les pages « Catégorie » téléchargées, nous utilisons *BeautifulSoup* pour analyser le code HTML et identifier les URL des produits.
4. Extraction des données : Une fois que nous avons extrait les URL des produits, nous répétons les deuxième et troisième étapes pour obtenir les éléments souhaités de chaque produit, tels que les noms, prix, descriptions, images, etc. Nous extrayons les informations souhaitées en sélectionnant les balises HTML pertinentes.
5. Stockage des données : Les données extraites sont ensuite stockées dans un format structuré tel que CSV, JSON ou directement dans une base de données via *DataGrip*. Cette dernière option permet d'utiliser le langage PostgreSQL pour interroger rapidement la base de données.
6. Gestion des limites et des interdictions : Nous faisons attention aux politiques d'utilisation de *24s* et mettons en place des mécanismes pour gérer les limites de fréquence des requêtes afin d'éviter d'être bloqués.

Afin d'économiser du temps de crawling et de s'adapter à un plus grand

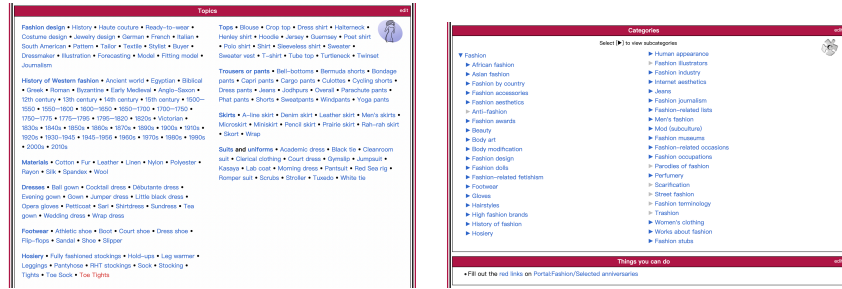


FIG. 4.1 : La structure hiérarchique des sujets et des catégories organisé par le portail « Mode » de Wikipedia.

nombre de sites de commerce électronique, nous n’avons pas immédiatement analysé les attributs et les informations spécifiques des produits lors de la 4ème étape, l’“extraction des données”. Au lieu de cela, dans notre base de données, nous avons temporairement stocké uniquement les trois colonnes : “id”, “info” (contenant les informations des produits au format JSON) et “images”. De cette façon, nous avons complété la collecte initiale des données produits brutes.

#### 4.1.2.2 Contenu informatif

Le domaine de la mode est riche en contenu informatif sur les tendances, les créateurs et les aspects culturels. Ces ressources offrent des données précieuses pour l’alignement des *LLMs*, tant sur le plan linguistique que linguo-visuel. Nous avons classé et organisé ces contenus en plusieurs catégories :

- **Les magazines de mode** : Publications comme *Vogue* et *Elle*, ainsi que des blogs spécialisés et des articles d’analyse sur les tendances actuelles et les créateurs influents. Ces articles peuvent être utilisés pour l’entraînement des *LLMs* afin de renforcer leur connaissance linguistique de base. Cet alignement linguistique aide les *LLMs* à comprendre et à générer le langage de la mode. En outre, les images peuvent être utilisées pour entraîner de grands modèles visuels. Il est important de noter que l’alignement visuel-linguistique est plus complexe que l’alignement unimodal en raison de la grande quantité de contenu textuel associé à chaque image.
- **Wikipedia** : L’encyclopédie en ligne collaborative offre une mine d’informations pour l’exploration et la fouille de texte. La plateforme Wikipedia comprend plusieurs ressources précieuses : les articles encyclopédiques qui

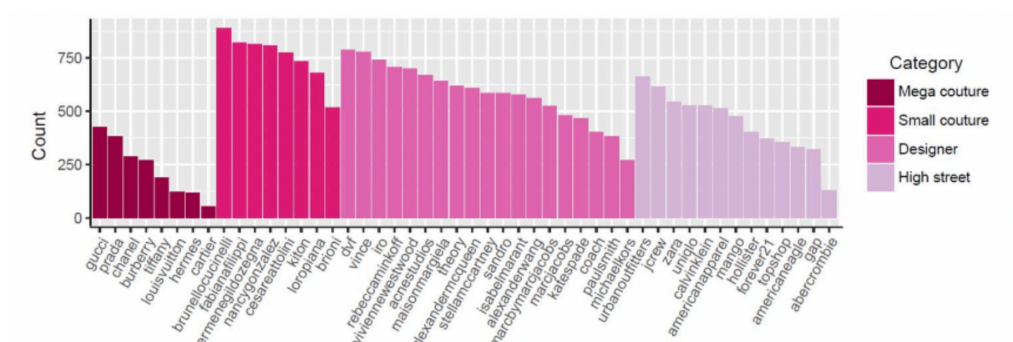


FIG. 4.2 : Fréquences de posts des tops 48 marques de la mode sur Instagram (HA et al. 2017).

couvrent une vaste gamme de sujets, le portail d'actualités qui offre les dernières nouvelles, WikiData qui fournit une base de connaissances structurée, et WikiBooks qui propose une collection de livres, y compris ceux sur la mode. Ces différentes branches de Wikipedia sont des sources inestimables pour l'acquisition et l'analyse de données textuelles dans le domaine de la mode (voir la figure 4.1).

- **Médias sociaux** : Dans le domaine de la mode, les médias sociaux englobent des plateformes variées telles que les blogs (comme Medium), les microblogs (comme Twitter), les forums (comme Reddit et Discord), les publications visuelles (comme Instagram et Pinterest), et les vidéos (comme YouTube et TikTok). Comparées aux ressources traditionnelles, ces plateformes sont plus dynamiques et favorisent une expression plus subjective et libre (voir la figure 4.2). Les textes et les commentaires présents sur ces réseaux sociaux offrent une grande diversité d'échantillons linguistiques, ce qui est extrêmement bénéfique pour l'entraînement des *LLMs*.

#### 4.1.2.3 Ensembles de données publics

Les sources de données mentionnées ci-dessus nécessitent toutes d'être collectées et organisées à l'aide de programmes de crawling. Afin de réduire la charge de travail du crawling, nous utilisons également certains ensembles de données publics pour enrichir notre corpus. Nous les classons en deux catégories principales : l'une dédiée au traitement automatique du langage (TAL) et l'autre à la multimodalité.

- **Pour le TAL** : En général, ces ensembles de données concernent les informations sur les produits, les commentaires des utilisateurs et le dialogue de service client.
  - **MMD Dataset**<sup>9</sup> (A. SAHA, KHAPRA et SANKARANARAYANAN 2018) offre une multitude de dialogues de type jeu de rôle de commerce électronique, où un individu joue le rôle du client et un autre celui du vendeur. En plus de cela, *MMD* publie également des informations sur les produits (environ un million de produits), y compris des images de produits, diverses propriétés, des avis des utilisateurs, etc. Les chercheurs de *MMD* ont également invité certains experts de la mode à fournir des informations complémentaires sur ces produits, notamment en fournissant une taxonomie simple, des astuces vestimentaires, ainsi qu'en compilant les mots utilisés pour les marques, les couleurs, les matériaux, etc. Ces travaux spécialisés seront utiles pour notre travail ultérieur sur la terminologie et la création de graphiques de connaissances. La partie des avis des utilisateurs et des dialogues simulés sur les produits peut servir de corpus d'entraînement pour les *LLMs*. Les informations sur les produits seront utilisées pour enrichir notre catalogue dans la section 4.2.
  - **Net-a-Porter Fashion Dataset**<sup>10</sup> contient des listes de produits de *Net-a-Porter* et de *Mr Porter*, deux détaillants de mode en ligne haut de gamme, pour femme et homme respectivement (environ 43k enregistrements). Les données, extraites de leurs sites web via Python et *BeautifulSoup*, comprennent quatre colonnes : la marque, la description du produit, le prix en USD et la catégorie du produit.
  - **Luxury Apparel Data**<sup>11</sup> contient les informations d'environ 5k produits. De manière similaire au jeu de données mentionné ci-dessus, il présente quatre colonnes : le nom du produit, sa description, sa catégorie et sa sous-catégorie.
  - **Women Clothing Reviews**<sup>12</sup> contient 23k commentaires sur les produits. Bien que le jeu de données ne présente pas le nom et d'autres

---

<sup>9</sup><https://amritasaha1812.github.io/MMD/>

<sup>10</sup><https://www.kaggle.com/datasets/justinpakzad/net-a-portermr-porter-fashion-dataset/>

<sup>11</sup><https://www.kaggle.com/datasets/chitwanmanchanda/luxury-apparel-data>

<sup>12</sup><https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews>



attributs spécifiques de chaque produit, il fournit une mini-taxonomie du produit ainsi que les notes et avis des utilisateurs à son sujet. Ces avis peuvent non seulement enrichir le corpus de l’entraînement pour les *LLMs*, mais aussi servir à des analyses de sentiment.

- **Fashion Conv Insta Dataset** (HA et al. 2017) contient de nombreuses images provenant d’Instagram, accompagnées de hashtags, de légendes et de scores de sentiment, ce qui constitue un vaste corpus pour enrichir notre terminologie et analyser le contexte du domaine de la mode (voir la figure 4.2).
- **Pour la multi-modalité** : dans cette partie, nous collectons des données visuelles dans le but d’aligner le langage sur les images, afin de renforcer le sens de la mode et la compréhension des attributs des produits par notre modèle. L’annotation fine associant chaque attribut de produit à une région précise des images est difficile à obtenir, donc la majorité des images dans les ensembles de données suivants ne disposent pas de cadres de délimitation (*bounding box*) au niveau des attributs.
  - **FashionViL** (Xiao HAN et al. 2022) a proposé un nouveau cadre d’apprentissage de la représentation V+L axé sur la mode en explorant l’apprentissage contrastif multi-vue pour plusieurs images d’un même produit et en alignant des concepts riches et détaillés visuellement et textuellement. À cette fin, il ont organisé et groupé cinq jeux de données publiques, à savoir *BigFACAD* (X. YANG et al. 2020), *FashionGen* (ROSTAMZADEH et al. 2018), *FashionzooK* (Xintong HAN et al. 2017), *FashionIQ* (H. WU et al. 2021a) et *PolyvoreOutfits* (VASILEVA et al. 2018). Ces jeux de données offrent 375k produits et plus d’un million de paires image-texte.
  - **FashionAI**<sup>13</sup> concentre à organiser et abstraire les étiquettes des attributs des vêtements pour construire un système de connaissances à la fois visuel et textuel en chinois.
  - **DeepFashion**<sup>14</sup> et **MMFashion**<sup>15</sup> offrent une gamme étendue de fonctionnalités pour l’analyse de la mode, couvrant la prédiction des attributs, la récupération des images par requête textuelle, la détection

<sup>13</sup><https://tianchi.aliyun.com/dataset/136948>

<sup>14</sup><https://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html>

<sup>15</sup><https://github.com/open-mmlab/mmfashion/>

des points de repères des vêtements, l'analyse et la segmentation des produits, la compatibilité et la recommandation des produits, ainsi que l'essayage virtuel de la mode.

- **GLAMI-1M** (KOSAR et al. 2022) contient plus d'un million de produits ainsi que leurs images et leurs textes dans 13 langues, ce qui en fait le jeu de données le plus vaste dans ce domaine.
- **MMD Dataset** (A. SAHA, KHAPRA et SANKARANARAYANAN 2018) propose non seulement de nombreuses données textuelles, mais il publie également environ 1 million de produits, avec en moyenne environ 4 images de différents angles pour chaque produit, accompagnées de leur nom, formant ainsi un corpus pour la recherche d'images par texte au niveau de l'image et l'alignement texte-image.

En plus, nous avons également consulté des jeux de données des différentes tâches e-commerciales tel que le comportement client, l'avis client, les requêtes historiques, etc, comme les suivants :

- **Amazon-M2** (W. JIN et al. 2024) : un jeu de données complet des sessions utilisateurs sur Amazon, contenant les clics historiques ainsi que le clic actuel, enrichi de métadonnées détaillées sur les produits.
- **Amazon Reviews 2023** (HOU et al. 2024) : plus de 500 millions d'avis utilisateurs sur Amazon, accompagnés de vérifications d'achats, couvrant 33 catégories de produits.
- **ECInstruct** (Bo PENG et al. 2024) : 116 528 échantillons dérivés de 10 tâches de commerce électronique exécutées à travers quatre catégories.
- **ESCI-data** (REDDY et al. 2022) : paires de requêtes d'achat et de résultats pertinents, complétées par des jugements de pertinence ESCI (Exact, Substitut, Complément, Non pertinent) pour évaluer la pertinence des produits.

## 4.2 Construction de la base de connaissance

Dans le texte ci-dessus, nous avons présenté nos méthodes d'acquisition de données brutes. Dans cette section, nous allons expliquer notre démarche de la construction de la base de connaissance, y compris comment organiser les données issues du commerce électronique en catalogues, comment extraire les termes pour créer des dictionnaires d'attributs, des taxonomies, ainsi que la

## 4.2 Construction de la base de connaissance

Dataset	Items	Imgs	Features	Langs
GLAMI-1M	1.11M	968k	image, name, description, class (191)	13
FACAD [13]	130k	993K	image, description, class (78)	1
Fashion-MMT [14]	110k	853k	image, description with noisy translations, class (78), attributes	2
Fashion550k [15]	550k	408k	image (in-the-wild), user comments, garment class, attributes, other metadata	1
Neti-look [16]	350k	355k	image (in-the-wild), comments	1
FashionGen [17]	78k	325k	image, description, class (121)	1
Amazon Fashion Products 2020 [18]	132k	132k+	multiple images, name, other	1
Fashion IQ [19]	50k	50k	image, description, attributes, relative caption	1
Fashion Product Images [20]	44k	44k	image, name, description, class, other	1

FIG. 4.3 : Aperçu des jeux de données et leurs features (KOSAR et al. 2022).

formalisation des connaissances contextuelles en embeddings pour la technique RAG (Retrieval-Augmented Generation) des LLMs.

### 4.2.1 Catalogue

Dans la section précédente, nous avons recueilli des données brutes sur les produits à partir de divers sites de commerce électronique. Afin de gagner du temps et de simplifier le processus de collecte, nous n'avons pas immédiatement analysé les attributs et les informations spécifiques de chaque produit. À la place, nous avons stocké toutes ces informations sans les organiser, au format JSON dans la colonne "info" de notre base de données. Dans cette section, afin de faciliter la consultation et la récupération des données ultérieurement, nous avons divisé toutes les informations d'un produit en six aspects : ID (comprenant spuID, skuID, etc.), description (comprenant le titre, la description, etc.), catégorie (comprenant la marque, le genre, la catégorie, la sous-catégorie, etc.), attributs (comprenant la couleur, le matériau, etc.), images (comprenant l'image principale, les images de détails, etc.) et prix (comprenant le prix, la devise, la remise, etc.).

Concernant la « catégorie » et les « attributs », même pour un même domaine (comme les montres ou les robes), la structure et les termes utilisés peuvent varier d'un site à l'autre. Par conséquent, nous ne cherchons pas immédiatement à fusionner les informations des produits de tous les sites, mais conservons les termes propres à chaque site. Pour les attributs, nous avons standardisé les données collectées en utilisant le format JSON suivant :

[

```
    {"attr_name": "color", "attr_value": "white"},  
    {"attr_name": "material", "attr_value": "cotton"}  
]
```

Ce format structuré est exploitable par des modèles de type RAG pour récupérer efficacement les informations pertinentes lors de la génération de texte.

#### 4.2.2 Terminologie et taxonomie

Chaque secteur possède ses propres termes et expressions spécifiques, et le domaine de la mode ne fait pas exception. Parallèlement, différentes marques ou plateformes utilisent également des terminologies variées, certaines allant même jusqu'à enregistrer des droits de propriété intellectuelle sur leurs termes exclusifs (par exemple, le terme "Louboutin" désignant les chaussures à semelles rouges exclusives de la marque Christian Louboutin). En outre, l'industrie de la mode est intrinsèquement liée à l'esthétique, à l'art, au commerce et même à l'histoire, et les termes spécialisés servent de mise en correspondance pour relier ces domaines. En organisant et en structurant ces terminologies, nous pouvons non seulement obtenir une vision d'ensemble plus précise, mais aussi établir des liens pertinents entre les différentes entités de ce domaine, y compris les synonymes, les antonymes et les relations hiérarchiques. Ainsi, la terminologie est cruciale pour la création ultérieure d'un système de dialogue basé sur la connaissance.

Dans notre travail, nous avons constitué une liste des marques et une taxonomie de termes, ainsi que les attributs et valeurs courants par domaine. Voici les étapes spécifiques :

1. Liste de marques : Nous avons compilé une liste la plus exhaustive possible des marques présentes dans le domaine de la mode. Cela inclut à la fois des marques de renommée internationale et des marques émergentes. Nous avons d'abord extrait une liste de marques unique à partir des catalogues obtenus dans la section précédente. Cette étape consistait à nettoyer les données et à éliminer les doublons pour obtenir une liste fiable. Ensuite, nous avons consulté diverses sources externes pour enrichir cette liste. Ces sources comprenaient des encyclopédies en ligne telles que *Wikipedia*, des médias spécialisés dans la mode et les affaires comme *Vogue* et *GQ*, et des annuaires commerciaux internationaux comme *Spex*. En complément, nous avons également vérifié les informations sur les marques de produits dans

des ensembles de données publics, tels que le *MMD*. En regroupant toutes ces informations, nous avons constitué une liste exhaustive et complète de marques de mode. Cette liste nous aide à normaliser les références aux marques dans notre base de données et à comprendre les préférences des consommateurs.

2. Catégorisation / Taxonomie : Nous avons développé une taxonomie détaillée des produits de mode, en classant les articles selon diverses catégories et sous-catégories. L'un des défis majeurs dans la création d'une taxonomie est l'hétérogénéité des classifications utilisées par les différents sites de commerce électronique. Chaque site peut adopter une structure de catégories unique, ce qui rend la normalisation complexe. Par exemple, la classification d'une jupe en jean peut varier significativement d'un site à l'autre. Sur un site A, la hiérarchie pourrait être : "vêtements pour femmes > jeans > jupes courtes, jupes en jean". Alors que sur un site B, la structure pourrait être : "vêtements pour femmes > jupes courtes > jupes en jean". Une autre variation courante est que certains sites peuvent inclure des catégories plus spécifiques dès le niveau supérieur, comme "vêtements pour femmes > jupes en jean, jupes courtes en jean".

Pour créer une taxonomie cohérente et utilisable, nous avons analysé les différentes structures de catégories et essayé de trouver des points communs et des divergences. Nous avons d'abord consulté le jeu de données *MMD*, qui déclare avoir regroupé les opinions d'experts de la mode pour définir leur taxonomie. En parallèle, grâce à la couverture exhaustive des produits de *Shein* et à son API ouverte<sup>16</sup>, nous avons pu obtenir directement la liste des catégories de *Shein*. Cette liste détaillait des informations telles que les identifiants de catégorie, les noms de catégorie, les identifiants parents et les listes de sous-catégories, ce qui s'est avéré extrêmement utile pour notre travail. Nous avons ensuite résumé ces travaux et obtenu une hiérarchie harmonisée qui permet de regrouper les produits de manière logique et intuitive, tout en tenant compte des variations observées.

3. Attributs et valeurs communes : Pour chaque domaine spécifique de la mode, nous avons minutieusement identifié les attributs les plus couramment utilisés ainsi que leurs valeurs correspondantes. Par exemple, pour les

---

<sup>16</sup><https://open.shein.com/documents/apidoc/detail/3000175-1000001>

CAT	EN	FR	CN	DEF-EN	DEF-FR	DEF-CN	SOURCE-EN	SOURCE-FR	SOURCE-CN
quality(metal)	10k	Or 10k	10k金	A gold alloy that includes 41.7% pure gold and 58.3% alloy	L'or 18K est constitué de 75% d'or pur, l'or 14K, de 58.5% et l'or 10K, à près de 42%	10k金则表示黄金含量为41.70%	<a href="https://www.josephjewellery.com/guide/glossary">https://www.josephjewellery.com/guide/glossary</a>	<a href="https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat">https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat</a>	<a href="https://www.blove.com/news/newatem-11113.html">https://www.blove.com/news/newatem-11113.html</a> <a href="https://www.luxurywatcher.com/zh-hans/article/26684">https://www.luxurywatcher.com/zh-hans/article/26684</a>
quality(metal)	14k	Or 14k	14k金	A gold alloy that includes 58.5% pure gold and 41.5% alloy	L'or 18K est constitué de 75% d'or pur, l'or 14K, de 58.5% et l'or 10K, à près de 42%	14k金则表示黄金含量为58.5%	<a href="https://www.josephjewellery.com/guide/glossary">https://www.josephjewellery.com/guide/glossary</a>	<a href="https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat">https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat</a>	<a href="https://www.blove.com/news/newatem-11113.html">https://www.blove.com/news/newatem-11113.html</a> <a href="https://www.luxurywatcher.com/zh-hans/article/26684">https://www.luxurywatcher.com/zh-hans/article/26684</a>
quality(metal)	18k	Or 18k	18k金	A gold alloy that includes 75% pure gold and 25% alloy	L'or 18K est constitué de 75% d'or pur, l'or 14K, de 58.5% et l'or 10K, à près de 42%	18K则表示黄金含量为75%	<a href="https://www.josephjewellery.com/guide/glossary">https://www.josephjewellery.com/guide/glossary</a>	<a href="https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat">https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat</a>	<a href="https://www.blove.com/news/newatem-11113.html">https://www.blove.com/news/newatem-11113.html</a> <a href="https://www.luxurywatcher.com/zh-hans/article/26684">https://www.luxurywatcher.com/zh-hans/article/26684</a>
quality(metal)	24k	Or 24k	24k金	A gold or gold alloy that is over 99.95% pure gold	L'or pur à 100% n'existe pas mais dans l'industrie des métaux précieux, on dit d'un or 24K qu'il est pur.	24K则表示黄金含量为99.99%	<a href="https://www.josephjewellery.com/guide/glossary">https://www.josephjewellery.com/guide/glossary</a>	<a href="https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat">https://flammeenrose.com/blogs/nouvelles/lor-et-son-carat</a>	<a href="https://www.blove.com/news/newatem-11113.html">https://www.blove.com/news/newatem-11113.html</a> <a href="https://www.luxurywatcher.com/zh-hans/article/26684">https://www.luxurywatcher.com/zh-hans/article/26684</a>
quality(gemstone)	4 Cs/4Cs/Four Cs	4c/4 C's Framework	4C钻石品质标准	A universal grading system developed by GIA (Gemological Institute of America) to evaluate the quality of a diamond. The 4Cs are Carat, Color, Clarity and Cut.	Pour simplifier le système de classification des diamants, les 4C (taille, couleur, pureté, poids) ont été co-créés avec le GIA (Gemological Institute of America) et présentés au public par De Beers en tant que langage universel. La combinaison de ces facteurs détermine la rareté et la valeur relatives d'un diamant.	GIA 首创现今业内公认的钻石分级标准：颜色、净度、切工和克拉重量。4C 钻石品质标准已成为评估钻石品质的全球通用方法。	<a href="https://www.beechvandcompany.com/jewelry-lingo-glossary/">https://www.beechvandcompany.com/jewelry-lingo-glossary/</a>	<a href="https://www.debeers.fr/fr/fr/4cs.html">https://www.debeers.fr/fr/fr/4cs.html</a>	<a href="https://4cs.gia.edu/zh-cn/4c-NE9192N8B1E7N9F83NE5H93N81NE9184N8A9NE6N4D87NE9N7N8/">https://4cs.gia.edu/zh-cn/4c-NE9192N8B1E7N9F83NE5H93N81NE9184N8A9NE6N4D87NE9N7N8/</a>

FIG. 4.4 : Extrait de la terminologie des bijoux-joaillerie-montres

montres, nous avons référencé le glossaire officiel de la Fondation de la Haute Horlogerie<sup>17</sup> ; pour les bijoux, nous avons consulté Joseph Jewellery<sup>18</sup>. En plus d'examiner les catalogues obtenus directement et les jeux de données comme *MMD* et *FashionGen*, nous avons consolidé les noms et les valeurs des attributs. Pour traiter les synonymes et les termes anonymes, nous avons également consulté des thésaurus en ligne. Pour structurer la hiérarchie et les dépendances entre ces termes, nous nous sommes appuyés sur les travaux d'Helmut Felber (FELBER 1980), qui identifie deux types de relations essentielles dans la construction de systèmes conceptuels : les relations logiques ou ontologiques. Dans la première catégorie, nous avons intégré des relations telles que l'hyperonymie (par exemple, le vêtement par rapport à la robe) et la coordination (par exemple, la veste et le blouson). Quant aux relations ontologiques, nous avons pris en compte la relation partie-tout (comme la manche et le blouson) ainsi que la coordination (comme le col et la manche).

Nous présentons les exemples de nos travaux dans les figures 4.4 et 4.5.

Pour le domaine de l'horlogerie, nous avons constitué un lexique spécialisé couvrant 17 grandes catégories de termes techniques et esthétiques (voir le tableau 4.1). Ce lexique comprend au total 868 entrées, bien que certains termes puissent appartenir à plusieurs catégories (par exemple, Chiffres arabes relève à la fois du *display* et de l'*aesthetic*). Pour le domaine de la joaillerie, nous avons

<sup>17</sup><https://www.hautehorlogerie.org/en/watches-and-culture/encyclopaedia/glossary-of-watchmaking/>

<sup>18</sup><https://www.josephjewellery.com/guide/glossary>

## 4.2 Construction de la base de connaissance

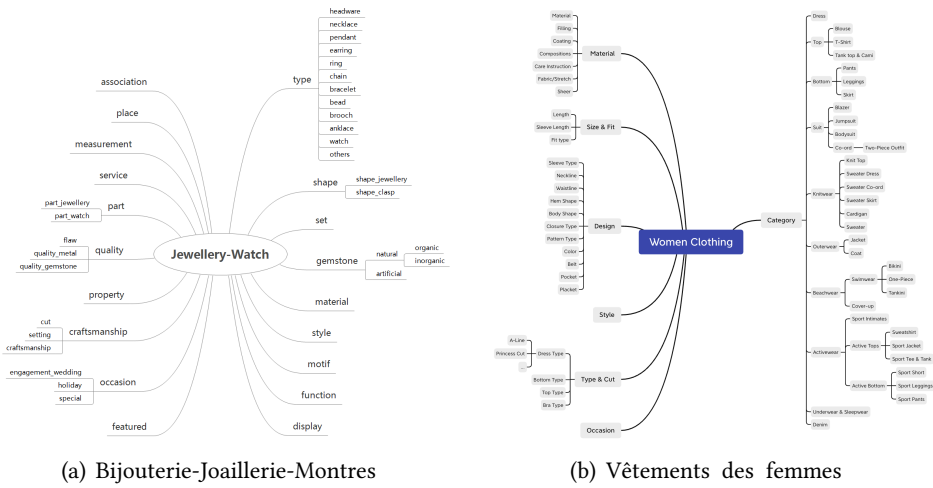


FIG. 4.5 : Structure des terminologies

défini 20 catégories, regroupant au total 1135 termes. En ce qui concerne les marques, certaines produisent à la fois des bijoux et des montres. Nous avons recensé au total 75 marques de renommée internationale.<sup>19</sup>

À partir des données organisées, nous avons initialement établi les relations de type "slot-value" qui seront utilisées dans notre système de dialogue (voir le tableau 4.2).

Comme montré dans le tableau 4.3, parmi les 48 services dans le domaine des vêtements femme, il contient en moyenne 12.5 *slots* par catégorie et dans chaque *slot*, il existe en moyenne 15.5 valeurs.

### 4.2.3 Connaissances contextuelles en embedding

Pour intégrer les connaissances de base du domaine de la mode dans notre système de dialogue, en plus des informations sur les produits, de la terminologie et de la taxonomie, nous avons également consolidé des articles sur l'esthétique, l'art et la culture comme connaissances supplémentaires (voir la section 4.1.2.2). Dans ce paragraphe, nous expliquons brièvement comment nous avons préparé ce contenu informatif en embeddings pour faciliter l'utilisation future de la technologie RAG (Retrieval-Augmented Generation) par les LLMs afin d'enrichir

<sup>19</sup>Le classement des marques et des gammes de prix s'appuie sur les informations disponibles sur le site spécialisé <https://www.xbiao.com/brand/>.

Catégorie	Nombre d'entrées	Exemples
parts	322	<i>Fond, Coussinet, Platine, Modérateur, ...</i>
times	49	<i>Double fuseau, Éphéméride, Horloge atomique, Heure d'été, ...</i>
jewelry	149	<i>Baguette, Astérisme, Calibré, Habitude cristalline, Dichroïsme, ...</i>
functions	43	<i>Calendrier perpétuel, Lunette, Altitude, ...</i>
aesthetic	71	<i>Art déco, Angler ou chanfreiner, Galvanoplastie, Oignon (montre), Peinture sur émail, ...</i>
energy_supply	19	<i>Autoquartz, Manuel, Perpétuelle,...</i>
display	93	<i>Analogique, Asthmomètre, Guichet, Chiffres arabes, Électroluminescence, ...</i>

TAB. 4.1 : Exemples de 7 des 18 catégories du lexique horloger et répartition des entrées. Certains termes appartiennent à plusieurs catégories (par ex. *Chiffres arabes*).

Domaine	Slot_name	Slot_value
Ring	gemstone carat	<i>diamond, ruby, perl, ...</i> <non-categorical>
Dress	type neckline discount	<i>a-line, shirt, tank, ...</i> <i>turtle neck, square, v-neck, ...</i> <non-categorical>
Watch	winding case_material case_diameter	<i>automatic, manual, ...</i> <i>titanium, white gold, ...</i> <non-categorical>

TAB. 4.2 : Exemple des attributs des articles

les conversations. La technologie RAG permet de combiner la génération de texte avec la récupération de passages d'informations pertinentes à partir d'une base de données pré-indexée. Cela signifie que lorsque le modèle de dialogue reçoit une question, il peut non seulement générer une réponse basée sur ses connaissances internes, mais aussi intégrer des informations spécifiques et actualisées récupérées à partir de la base de données.

Dans notre cas, les informations sur les produits et les connaissances contextuelles (les contenus informatifs mentionnés ci-dessus) peuvent toutes être transformées en embeddings, puis utilisées par la technique RAG pour améliorer les capacités de recherche des LLMs. Étant donné que les informations sur nos produits sont déjà traitées sous forme de données structurées (catalogue, ter-



num. services	59
avg. num. slots per service	12.5625
avg. num. values per slot	15.5167

TAB. 4.3 : Aperçu des slots choisis

minologie, taxonomie, relations *slot-valeur*, etc.), et que la recherche dans des données structurées est relativement simple, notre travail consiste uniquement à transformer les connaissances contextuelles en embeddings. Pour sélectionner et fine-tuner un modèle d'embedding adapté à notre domaine, nous avons effectué les étapes suivantes :

1. Préparation d'un ensemble de données de paires requête-contexte : nous avons utilisé les articles, les pages Wikipedia et autres contenus informatifs collectés dans les chapitres précédents (voir la section 4.1.2.2) comme matériaux de base. Pour chaque contenu, nous avons employé GPT-3.5 pour extraire des paires requête-contexte pertinentes. Concrètement, nous avons fourni à GPT-3.5 des exemples de paires, tels que :

```
[
  {"query": "Pourquoi Hermès est-il si cher ?",
   "context": "Hermès est une marque de luxe fondée en 1837,
réputée pour ses produits faits à la main..."},
  {"query": "What is the most popular luxury brand?",
   "context": "The brand Prada is very popular among gen-Z..."}
]
```

Cette approche est inspirée d'un jeu de données intitulé « Financial-QA »<sup>20</sup>, qui contient des paires de question-réponse-contexte dans le domaine financier. Puisque nous avons besoin uniquement d'obtenir un modèle d'embeddings pour la requête, nous avons réduit les éléments question-réponse-contexte en requête-contexte. Ainsi, nous avons systématiquement généré des paires requête-contexte à partir de nos données informatives pour l'entraînement suivant.

2. Évaluation du Modèle Pré-entraîné : Nous avons sélectionné certains modèles pré-entraînés sur *SentenceTransformer* et les avons évalués avec l'outil

<sup>20</sup><https://huggingface.co/datasets/viratth/financial-qa-10K>

*InformationEvaluator* en utilisant la métrique *NDCG* (*Normalized Discounted Cumulative Gain*) sur notre ensemble de données pour établir une base de référence. La métrique *NDCG* donne une indication de la qualité de la requête par la position du document pertinent dans le classement. Elle prend en compte la position du document pertinent dans le classement, ce qui signifie que les documents pertinents sont plus importants s'ils sont classés plus haut.

3. Fine-tuning du Modèle d'Embedding : pour optimiser le modèle, nous définissons la fonction de perte en tant que *Matryoshka Representation*, qui est conçue pour affiner les embeddings de manière à capturer des relations complexes et pertinentes au sein des données. Ensuite, nous utilisons l'outil *SentenceTransformersTrainer* pour effectuer le fine-tuning.
4. Création des Embeddings : nous avons utilisé des modèles fine-tunés pour créer des embeddings vectoriels de chaque article, en faisant attention à segmenter les articles en morceaux de longueur similaire à celle des contextes dans l'ensemble de données des paires requête-contexte. Ces embeddings permettent de représenter le contenu de manière compacte et d'effectuer des recherches sémantiques efficaces. Les modèles utilisés pour cette tâche sont les mêmes que ceux qui seront employés pour le système de dialogue, assurant ainsi une cohérence dans le traitement des données.
5. Indexation et stockage : nous avons utilisé des outils tels qu'*Annoy* pour indexer et stocker les embeddings générés. *Annoy* est particulièrement efficace pour les recherches de texte intégral et les requêtes complexes, ce qui assure un accès rapide et précis aux informations pertinentes lorsque le modèle de dialogue en aura besoin.

Ainsi, notre base de connaissances sous forme d'embeddings est créée.

#### 4.2.4 *Persona utilisateur*

Un persona utilisateur est une représentation fictive et détaillée d'un type d'utilisateur cible, basée sur des données réelles et des caractéristiques démographiques, comportementales et psychographiques. Les personas sont utilisés pour comprendre les besoins, les motivations et les comportements des utilisateurs, permettant ainsi de concevoir des expériences utilisateur plus personnalisées et efficaces.

Les personas peuvent être structurés ou non structurés. Les personas structurés incluent des traits précis tels que le genre, l'âge, le métier, la zone de résidence, etc. Les personas non structurés, quant à eux, incluent des descriptions plus libres et qualitatives, telles que « J'aime le sport, je travaille beaucoup et je préfère les vêtements confortables ». À partir des personas structurés, nous pouvons générer des personas non structurés, par exemple, « Je suis une femme de 34 ans, vivant à Paris, travaillant comme graphiste, et intéressée par la mode durable ».

Dans notre étude, nous avons exploré deux travaux notables sur la création de personas :

Premièrement, le jeu de données « Fashion-Style-Instruct »<sup>21</sup> qui décrit les besoins vestimentaires des utilisateurs en fonction de leur morphologie et de leur style personnel ainsi que du contexte de l'événement. Par exemple, une personne pourrait se décrire comme ayant un « corps en forme de poire qui adore les tendances féminines » et demandant des vêtements pour un rendez-vous pique-nique. Ce jeu de données contient le persona utilisateur, le contexte de l'événement, ainsi que la réponse de l'agent conversationnel (des propositions de vêtements générées par ChatGPT). Cette approche permet de créer des personas détaillés basés sur des caractéristiques physiques et des préférences de style spécifiques.

Deuxièmement, dans le jeu de données « PersonaChat Maxi » (MAZARÉ et al. 2018), les chercheurs ont extrait des personas à partir des commentaires des utilisateurs de *Reddit*. L'objectif était de générer une multitude de personas pour des agents de dialogue en domaine ouvert. Contrairement à notre cas d'utilisation orienté vers l'achat de vêtements et d'accessoires, ce travail se concentre sur les conversations en domaine ouvert.

Pour notre projet, en regroupant les méthodes ci-dessus, nous avons structuré les personas utilisateurs de manière à répondre spécifiquement aux besoins du commerce électronique de mode. Chaque persona est divisé en trois aspects principaux :

1. **Informations de base** : âge, sexe, zone de résidence, emplacement, statut social, revenu et intérêts. Ces informations fournissent un contexte général sur l'utilisateur.
2. **Objectif d'achat** : article, taille, détails, style préféré, caractéristiques phy-

---

<sup>21</sup><https://huggingface.co/datasets/neuralwork/fashion-style-instruct>

siques, contexte de l'événement, etc. Ce segment se concentre sur les intentions spécifiques de l'utilisateur en matière d'achat.

3. **Style de conversation et personnalité** : patience, réceptivité, demandes, etc.  
Cet aspect vise à personnaliser l'interaction en fonction des préférences de communication de l'utilisateur.

Pour chaque aspect et chaque élément, nous avons défini des valeurs candidates et créé des personas structurés pour la première étape (voir le tableau 4.4). Ensuite, nous les avons élaborés à l'aide de modèles et de ChatGPT pour obtenir des personas en langue naturelle.

En intégrant ces éléments, nous avons pu créer des personas utilisateurs riches et adaptés, permettant ainsi une meilleure personnalisation de l'expérience d'achat en ligne.

TAB. 4.4 : Description des éléments des personas clients

Element	Explication
Age	Tranche d'âge du client
Gender	Homme, femme, ou LGBTQIA+
Living Area	Le pays ou la région de résidence
Interest	Intérêts du client dans la vie quotidienne
Location	Urbain, périurbain ou rural
Social Status	Profession et niveau de revenu
Item	Les produits ciblés par le client
Size	Taille de vêtement appropriée pour le client
Style	Le style personnel préféré du client
Physic	Les caractéristiques physiques du client
Occasion	Le contexte de l'événement
Details	Conception détaillée des produits ciblés
Patience	Niveau de patience, indiqué par le nombre de questions dans chaque tour
Receptiveness	Comment le client répond aux conseils de l'assistant de magasin
Demand (optional)	Niveau d'exigence en matière d'achat et si le client est bavard

4.3 Génération de données de dialogue

Pour entraîner un agent conversationnel, les données de dialogue jouent un rôle crucial. Dans cette section, nous allons examiner les différentes méthodes pour obtenir les données de dialogue et les comparer.

#### 4.3.1 *Différentes méthodes pour obtenir les données de dialogue : collecte ou génération*

Dans cette partie, nous introduisons concrètement les différentes méthodes pour obtenir un jeu de données de dialogue *TOD*, puis analysons les avantages et les inconvénients de ces méthodes. Généralement, nous divisons ces méthodes en deux approches : collecte ou génération, en fonction du degré de sollicitation d'humains ou de machines.

##### 4.3.1.1 *Collecte des dialogues*

La collecte de véritables dialogues entre êtres humains est une approche courante dans les scénarios de conversations à domaine ouvert (*OOD*) (FENG et al. 2022). Par contre, pour collecter de véritables conversations *TOD*, il serait nécessaire que les entreprises partagent, le moyen le plus direct est que les entreprises divulguent les conversations réelles et désensibilisées entre le service client et les clients (M. CHEN et al. 2020 ; N. DU et al. 2019). Cette étape demande souvent un travail intensif d'annotation humaine. L'avantage de cette méthode est que toutes les conversations collectées ont réellement eu lieu, ce qui garantit les vraies intentions et les expressions naturelles dans les scénarios, apportant ainsi un jeu d'entraînement assez adéquat du point de vue de la distribution des données. Par contre, un des prérequis pour cette méthode est d'avoir déjà une entreprise ou une plateforme capable de collecter ces dialogues réels, ce qui est rare et soulève des problèmes de confidentialité.

Afin de remédier à ces difficultés, certains chercheurs qui n'ont pas accès à ces dialogues essayent de collecter les commentaires ou des questions-réponses publics sur les sites internet, pour les transformer en format de dialogues, par exemple *InstaConv* (HA et al. 2017), *YelpNLG* (ORABY et al. 2019), et d'autres (M. HENDERSON, VULIC et al. 2019 ; JI et al. 2020). Cependant, les conversations ne sont pas ancrées dans une base de connaissances existante ni d'interfaces de fonction (API), ce qui limite leur usage direct pour l'entraînement de systèmes véritablement orientés vers des tâches (BUDZIANOWSKI et al. 2018).

Outre la collecte des conversations et des commentaires réels en ligne, les chercheurs ont adopté une autre approche pour obtenir les conversations entre les êtres humains : le crowdsourcing. Cette méthode consiste à externaliser des tâches en faisant appel à de grands groupes de personnes, généralement via

- You are traveling to Cambridge and looking forward to try local restaurants.
- You are looking for a **place to stay**. The hotel should be in the type of **hotel** and should be in the **centre**.
- The hotel should **include free wifi** and should have a **star of 4**.
- Once you find the **hotel** you want to book it for **3 people** and **5 nights** starting from **monday**.
- Make sure you get the **reference number**.
- You are also looking for a **restaurant**. The restaurant should serve **australasian** food and should be in the **moderate** price range.
- The restaurant should be **in the same area as the hotel**.
- If there is no such restaurant, how about one that serves **british** food.
- Once you find the **restaurant** you want to book a table for **the same group of people** at **18:30** on **the same day**.
- Make sure you get the **reference number**

FIG. 4.6 : Exemple des directives *MultiWoZ* (BUDZIANOWSKI et al. 2018)

Internet. Le travail le plus représentatif est le Wizard-of-Oz (WoZ, J., KELLEY et WATSON 1984). Le cadre WoZ a été initialement proposé comme une approche itérative pour améliorer les expériences utilisateur lors de la conception d'un système conversationnel. L'objectif de la collecte de données WoZ est d'enregistrer des conversations pour le développement futur du système. En modifiant le cadre WoZ d'origine pour le rendre adapté au crowdsourcing, une série de travaux ont été effectués via la plateforme *AMAZON Mechanical Turk* (voir la section 2.2). Pour obtenir les dialogues simulés entre les êtres humains, les chercheurs invitent des participants, leur fournissent des scénarios et des objectifs spécifiques, les incitant ainsi à dialoguer selon diverses directives prédéterminées (voir la figure 4.6).

Récemment, davantage de jeux de données de dialogues multimodaux ont également été créés par ces méthodes, dont les sujets varient de la navigation à la compréhension des photos de produits, comme nous l'avons introduit dans la section 2.2.

## 4.3.1.2 Génération des dialogues

En sachant que la collecte de dialogues humain-humain est toujours coûteuse, la méthode de génération/simulation est considérée comme une alternative intéressante par les chercheurs. L'idée principale est d'utiliser partiellement ou entièrement des modèles pour créer les simulateurs et les annotations.

**Dialogues humain-machine** Étant donné que construire un corpus de dialogues *TOD* à partir de zéro est difficile, la plupart des corpus de ces dialogues sont développés sur la base d'un système de dialogue existant. Un exemple célèbre de ce type est le système d'information sur les bus *Let's Go*, qui fournit en direct des informations sur les horaires de bus par téléphone (RAUX et al. 2005). Ce travail a conduit à la première édition du Défi de Suivi de l'État de Dialogue (*DSTC*, J. WILLIAMS, RAUX et al. 2013). En développant l'idée du système *Let's Go*, les deuxième et troisième *DSTC* (M. HENDERSON, THOMSON et J. WILLIAMS 2014a,b) ont produit des ensembles de données humain-machine pour la recherche de restaurants dans la région de Cambridge, au Royaume-Uni. Depuis lors, les *DSTC* sont devenus l'un des sujets centraux dans la communauté de la recherche sur le dialogue.

Dans le travail *bAbI* (BORDES et WESTON 2016), les chercheurs ont déployé un simulateur d'agent qui peut appeler les *API* de services d'après les besoins de l'utilisateur. On a invité les testeurs humains à dialoguer avec cet agent simulé et enregistré les interactions humain-machine. Bien que la collecte de données humain-machine soit une solution évidente pour le développement des systèmes de dialogue, elle n'est possible qu'avec la mise à disposition d'un système fonctionnel existant. Par conséquent, cela devient un problème du cercle vicieux.

Les données obtenues par ce moyen sont plutôt utilisées pour l'amélioration des systèmes existants que pour le développement de systèmes dans un tout nouveau domaine. Ce qui est encore plus problématique, c'est que la capacité du système initial introduit des biais supplémentaires dans les données collectées, ce qui peut entraîner un déséquilibre entre les ensembles d'entraînement et de test (T.-H. WEN, GASIC, MRKSIC, ROJAS-BARAHONA, P.-h. SU, ULTES et al. 2016). La capacité de compréhension limitée du système initial peut inciter les utilisateurs à s'adapter à des exemples d'entrée plus simples que le système peut comprendre,

mais qui ne sont pas nécessairement naturels dans les conversations. En même temps, on peut voir que l’effort humain dans ce genre de travail est encore important. On peut dire que cette méthode pour obtenir les dialogues humain-machine se situe à mi-chemin entre la collecte de dialogues et la génération de dialogues.

**Dialogues machine-machine** La création d’un environnement avec un utilisateur simulé permet de générer de manière exhaustive des templates de dialogues. Ces templates peuvent être transformés en langage naturel soit par des règles prédéfinies (BORDES et WESTON 2016), soit par crowdsourcing (SHAH et al. 2018). Cette approche garantit une diversité et une couverture complète de tous les résultats de dialogue possibles dans un domaine donné.

Suivant l’idée de simulateurs machine-machine (SHAH et al. 2018), certains chercheurs essayent de réduire davantage le travail humain en utilisant deux automates simulateurs comme les deux agents du dialogue, un pour le rôle de ”l’utilisateur” et l’autre pour celui de ”l’agent”. Après la génération des templates de conversations par les simulateurs, une étape de reformulation ou de mise en forme est appliquée, soit par des experts humains (RASTOGI et al. 2019 ; SHAH et al. 2018), soit par des modèles de langage (H. LEE et al. 2022 ; T. ZHAO et ESKÉNAZI 2018).

Ces dialogues générés ne sont pas nécessairement très naturels, car ils dépendent entièrement de la configuration prédéfinie des utilisateurs et des systèmes de robots. Cela peut les rendre rigides et limités, ce qui entraîne un risque de désaccord entre les données d’entraînement et les interactions réelles, nuisant ainsi à la qualité du système de dialogue entraîné.

**Dialogues entre LLMs** Étant donné les avantages et les défis des dialogues générés par les machines, avec le développement évident des capacités des grands modèles de langage, de plus en plus de chercheurs optent pour l’utilisation des LLMs tels que *ChatGPT* (ACHIAM et al. 2023 ; OUYANG et al. 2022), *LLaMA* (TOUVRON, LAVRIL et al. 2023 ; TOUVRON, MARTIN et al. 2023) ou *Alpaca* (TAORI et al. 2023) afin de générer des données d’entraînement pour diverses tâches, y compris des dialogues.

Au début, certains travaux tels que *ChatGPT* et *LLaMA* ont utilisé des données provenant de diverses sources telles que des conversations collectées sur



divers forums, *Wikipedia*, du code et des commentaires sur *GitHub*, des articles académiques, etc., pour entraîner un grand modèle de langage à partir de zéro. D'autres travaux, comme *Alpaca* pour le domaine ouvert et d'autres *LLMs* spécifiques à un domaine, ont commencé à utiliser de grands modèles pré-entraînés pour générer des données d'entraînement, puis à effectuer un *fine-tuning* ou un *prompt-tuning* dessus. Par exemple, *ShareGPT*<sup>22</sup>, *UltraChat* (DING et al. 2023), *GPTeacher*<sup>23</sup>, *MiniGPT4* (D. ZHU et al. 2023), etc.

Pour obtenir spécifiquement des dialogues, nous nous référons aux deux travaux suivants.

**PersonaChatGen** : une mise à jour de *PersonaChat* (Saizheng ZHANG et al. 2018). Il a été démontré que *GPT-3* peut générer des dialogues personnalisés en imitant diverses personnalités. Un *LLM* de *Blender* 90M entraîné sur cet ensemble de données a obtenu les meilleures performances (Y.-J. LEE, LIM et al. 2022).

**SalesBot** (CHIU et al. 2022) : ce jeu de données contient des dialogues, chacun commençant par des conversations sociales, suivies par la découverte d'une intention potentiellement orientée vers une tâche de l'utilisateur, et se terminant par l'accomplissement de cette tâche. Pour générer de tels dialogues, les auteurs suivent une approche en quatre étapes : la génération de dialogues *OOD*, la détection des intentions potentielles de l'utilisateur, la transition des dialogues *OOD* vers *TOD*, et la génération de dialogues *TOD*. Pour chaque étape, ils utilisent différents modèles de langage respectifs :

1. *OOD* : généré par deux *BlenderBots* (ROLLER et al. 2020) pré-entraînés sur l'ensemble de données *BST* (E. M. SMITH et al. 2020) et les personas correspondants.
2. Détection des intentions : réalisée en utilisant les systèmes de question-réponse (*QA*). Ils prennent l'énoncé de l'utilisateur comme contexte et posent une question binaire (oui-ou-non), permettant à un modèle de *QA* *DistillBert* (SANH et al. 2019) de répondre si le contexte implique une intention de l'utilisateur. Ce modèle de *QA* est d'abord pré-entraîné sur *SQuAD 2.0* (RAJPURKAR, JIA et P. LIANG 2018) puis affiné sur l'ensemble de données *SGD* (RASTOGI et al. 2020). La question binaire est générée par des modèles de templates et reformulée par un modèle *T5* (RAFFEL et al. 2019) entraîné sur *Quora Question Pairs* (Zhiguo WANG, HAMZA

<sup>22</sup><https://sharegpt.com/>

<sup>23</sup><https://github.com/teknium1/GPTeacher/tree/main>

et FLORIAN 2017).

3. Transition de *OOD* à *TOD* : générée par un *T5* entraîné sur *OTters* (SEVEGNANI et al. 2021).
4. *TOD* : généré de deux manières : 1) trouver quelques dialogues *TOD* pertinents avec la même intention dans l'ensemble de données *SGD*, mais en version délexicalisée (K. SUN et al. 2020) ; 2) laisser deux *BlenderBots TOD* dialoguer entre eux, l'un sur l'énoncé de l'utilisateur et l'autre sur l'énoncé du système, tous deux entraînés également sur *SGD*.

L'approche pour générer des dialogues et d'autres ensembles de données pour diverses tâches à l'aide de *LLMs* repose essentiellement sur l'utilisation des *LLMs* pré-entraînés, qui comprennent non seulement la capacité de compréhension et de génération de langage, mais aussi les vastes connaissances de fond acquises lors de la phase de pré-entraînement. Ces connaissances couvrent divers domaines tels que le bon sens, l'histoire, la science, et la culture, et proviennent de sources de données variées telles que *Wikipédia* et les documents de musées. Ainsi, cette approche permet de générer des données plus diversifiées, riches et de haute qualité, sans nécessiter l'intervention d'experts humains.

Bien sûr, cette méthode comporte également des inconvénients. Par exemple, elle est limitée par les capacités du *LLM*. Sans évaluation humaine, il est difficile de garantir que les résultats générés par le *LLM* sont entièrement corrects. De nombreux articles de recherche ont signalé des problèmes d'hallucinations dans les sorties de *LLM* (EVANS et al. 2021 ; OPENAI et al. 2024 ; Baolin PENG et al. 2023). De plus, certaines *API* de *LLM* sont payantes<sup>24</sup>. Enfin, l'utilisation de *LLM* de grande taille peut exiger des configurations matérielles plus puissantes.

Dans cette section, nous avons présenté plusieurs méthodes variées pour obtenir des données de dialogue. On peut constater qu'aucune méthode n'échappe au « triangle impossible » (voir la figure 4.7), c'est-à-dire qu'aucune méthode ne parvient à satisfaire simultanément les trois critères : « énoncés naturels », « coût bas » et « biais réduit ». « Énoncés naturels » signifie des expressions plus authentiques et semblables à celles d'un être humain, ce qui nécessite inévitablement une implication significative du travail humain, entraînant des coûts plus élevés. Même en utilisant un *LLM*, tels que l'*API* de *ChatGPT*, les coûts demeurent élevés. Les méthodes privilégiant l'utilisation de machines par

---

<sup>24</sup><https://openai.com/product>

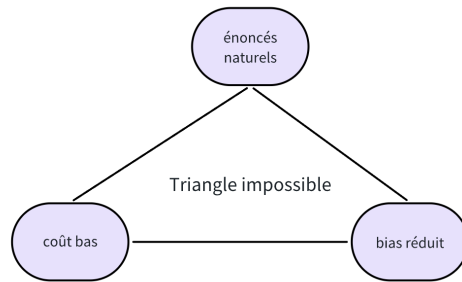


FIG. 4.7 : Le triangle impossible d'obtenir les données de dialogue

rapport au travail humain permettent de maîtriser les coûts, mais cela se fait au détriment de la naturalité des énoncés et de la qualité des dialogues. Par ailleurs, la prédétermination des scénarios peut être insuffisante, ce qui empêche de garantir une réduction significative des biais. Si l'on souhaite disposer de suffisamment de scénarios prédéfinis et réalistes, le travail de collecte doit nécessairement être plus conséquent que celui de la génération, entraînant une augmentation des coûts.

#### 4.3.2 Génération des dialogues par humain

Dans la section 4.1.2, nous avons mentionné quelques travaux antérieurs précieux, dont le jeu de données *MMD* (A. SAHA, KHAPRA et SANKARANARAYANAN 2018). En raison de sa date de création ancienne et du grand volume de dialogues qu'il contient, il présente de nombreux problèmes hérités du passé. Par exemple, un utilisateur cherche des « sandales en cuir », mais le vendeur recommande une série de montres, et la conversation se poursuit inexplicablement ; ou encore, lorsque le titre du produit est inséré de manière artificielle dans le dialogue, cela crée des phrases incohérentes. Ces problèmes peuvent tous être résolus avec l'aide de *LLM*. Nous avons sélectionné 1K dialogues dans ce jeu de données et les avons corrigés manuellement comme un jeu de données de dialogue de référence (en anglais, *golden standard*).

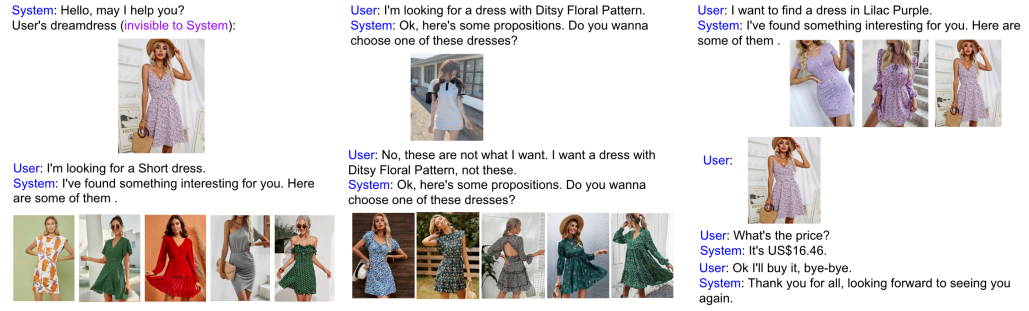


FIG. 4.8 : Illustration d'un dialogue multimodal simulé.

### 4.3.3 Génération de dialogues par les automates

Dans cette section, notre objectif est de simuler des conversations à l'aide des automates, comme illustré dans la figure 4.8.

Les automates, ou simulateurs formels sont largement utilisés dans l'agent conversationnel. Ici les simulateurs « formels » signifient que des systèmes informatiques sont programmés suivant des règles spécifiques au domaine et des règles linguistiques. Pour mieux cibler les dialogues que nous souhaitons utiliser, nous définissons le scénario comme étant les dialogues entre un client et un vendeur, et nous tentons de simuler leurs conversations et actions à l'aide de deux simulateurs distincts. Notre approche globale consiste à d'abord initialiser les personnages et les états des simulateurs pour les deux côtés respectivement, puis à permettre aux deux simulateurs de communiquer en utilisant des annotations en triplet *NLU*, et enfin à traiter ces annotations pour générer du langage naturel, voir la figure 4.9.

Dans la suite, nous prendrons l'exemple du scénario « achat d'une robe » pour illustrer comment nous initialisons les états initiaux de l'utilisateur *U* et du vendeur *V*, ainsi que comment nous définissons les règles de réaction entre eux, comment nous introduisons certains personnages pour rendre la situation plus variée, et comment nous remplissons le cadre du dialogue en langage naturel.

Nous utilisons un simulateur basé sur l'agenda pour modéliser le comportement de l'utilisateur *U*, tandis qu'un simulateur basé sur une machine à états finis est employé pour représenter le vendeur *V*. Faisant référence au travail précédent (SHAH et al. 2018), nous définissons le processus de dialogue comme une séquence de transitions d'états de deux côtés. Ici dans notre cas, à chaque action *a*, les

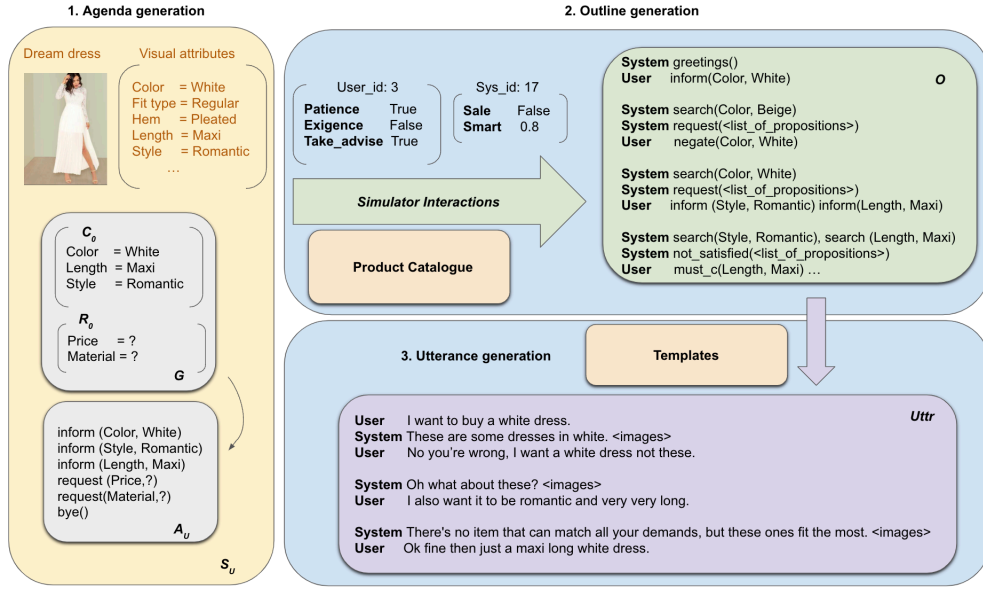


FIG. 4.9 : Le cadre de la génération de dialogue par simulateurs en 3 étapes.

états de  $U$  et  $V$  sont mis à jour comme montrés dans la formule 4.1 :

$$S_U^t \rightarrow a_V^t \rightarrow S_U'^t \rightarrow a_U^t \rightarrow S_U^{t+1},$$

$$S_V^t \rightarrow a_U^t \rightarrow a_V^t \quad (4.1)$$

$$\searrow S_V^{t+1} \quad (4.2)$$

L'état de l'utilisateur au tour  $t$ , noté  $S_U^t$ , change à un état intermédiaire  $S_U'^t$  après avoir reçu l'action du vendeur  $a_V^t$ . Avec la sélection de la réaction de l'utilisateur  $a_U^t$  à partir d'un ensemble d'actions possibles, l'état de l'utilisateur se met à jour en  $S_U^{t+1}$ . On considère le processus comme une chaîne de Markov, il est modélisé par trois modèles pour le côté user :  $P(S_U'^t | a_V^t, S_U^t)$  pour la première transition d'états,  $P(a_U^t)$  pour la sélection de l'action de l'utilisateur, et  $P(S_U^{t+1} | a_U^t, S_U'^t)$  pour la deuxième transition de l'état de l'utilisateur. De l'autre côté, le vendeur  $V$  prend son action dès qu'il reçoit l'action de l'utilisateur  $a_U^t$  pour répondre à l'utilisateur  $U$ , et son état  $S_V^t$  passe à  $S_V^{t+1}$ . On modélise le processus du vendeur par deux modèles :  $P(S_V^{t+1} | a_U^t, S_V^t)$  et  $P(a_V^t | a_U^t, S_V^t)$ .

#### 4.3.3.1 État, objectif et agenda

$$S = (A, G) \quad \text{et} \quad G = (C, R) \quad (4.3)$$

L'état  $S$  de chaque simulateur se compose de deux parties : un agenda  $A$  qui stocke les actions à venir et un ensemble d'objectifs  $G$  qui contient les informations sur les attributs des produits que les deux parties souhaitent connaître ou ont déjà connaissance. Cet ensemble d'objectifs  $G$  comprend également deux aspects : les contraintes ou les conditions connues  $C$  et les requêtes ou les résultats  $R$ . Du point de vue de l'utilisateur,  $C$  représente les attributs des produits qu'il souhaite que le produit recherché possède, par exemple s'il recherche une robe longue de couleur blanche, les attributs "couleur : blanche" et "longueur : longue" sont deux conditions à satisfaire. Quant à  $R$ , d'une part, il s'agit des attributs que l'utilisateur souhaite demander, par exemple le prix du produit, d'autre part, il s'agit des résultats d'une recherche que le vendeur obtient après avoir consulté la base de données. En résumé, nous pouvons décrire simplement la relation entre l'état  $S$ , l'ensemble d'objectifs  $G$  et l'agenda  $A$  comme la formule 4.3.

**Initialisation des états** Regardons d'abord le côté de l'utilisateur  $U$ . Les contraintes  $C$  et les requêtes  $R$  forment ensemble les objectifs  $G$  de l'utilisateur lors d'une session de dialogue, ce dernier détermine les sujets du dialogue entier. Nous avons divisé les attributs des produits en deux catégories : les attributs visuels et les attributs non visuels. Pour les attributs visuels, l'utilisateur peut les voir dans l'image, ce qui lui permet de juger si les attributs affichés sur l'image correspondent à ses contraintes, comme la couleur, la longueur, etc. En revanche, les attributs non visuels, tels que le prix, le matériau, l'imperméabilité, ne peuvent pas être directement confirmés à partir de l'image. Ainsi, les requêtes  $R$  sont limitées aux attributs non visuels, c'est-à-dire aux informations que l'utilisateur ne peut pas discerner d'un seul coup d'œil et qu'il doit demander au vendeur. Quant aux contraintes  $C$ , elles peuvent être visuelles ou non visuelles. Tant les contraintes que les requêtes sont basées sur les informations d'attributs de produit d'un catalogue de robes. Les contraintes  $C$  sont des conditions données, écrites sous la forme  $\{attr = value\}$ , tandis que les requêtes  $R$  sont des demandes, écrites sous la forme  $\{attr = ?\}$ .

L'ensemble  $A$  est un agenda qui contient les actions à entreprendre vis-à-vis des contraintes et requêtes dans l'ensemble  $G$ . Pour chaque contrainte  $c$ , l'action de l'utilisateur est « informer » (*inform*), tandis que pour chaque requête, l'action de l'utilisateur est « demander » (*request*). Enfin, pour l'utilisateur, nous ajoutons une dernière action « acheter » (*buy*) pour indiquer sa décision finale de prendre le produit recommandé ou non (voir la formule 4.5).

$$A_U = \text{inform}(C) + \text{request}(R) + \text{buy}() \quad (4.4)$$

Prenons l'exemple dessus, supposons que l'utilisateur  $U$  veut trouver une robe de rêve, cette robe serait mieux en couleur blanche et très très longue. Le vendeur  $V$  cherche dans la catalogue pour trouver les robes satisfaisantes ou similaires et  $U$  demande le prix de la robe puis décide s'il la prend ou non. L'agenda  $A_U$  sera donc le suivant :

$$\left[ \begin{array}{l} \text{inform}(\text{color} = \text{white}) \\ \text{inform}(\text{length} = \text{maxi}) \\ \text{request}(\text{price} = ?) \\ \text{buy}() \end{array} \right]$$

Du côté du vendeur  $V$ , ses objectifs  $G$  sont de recevoir progressivement les contraintes  $C$  de l'utilisateur et de trouver les produits dans le catalogue qui satisfont au maximum ces contraintes. Ensuite, il doit fournir les informations sur le produit demandé par l'utilisateur dans  $R$ . Au début de la conversation, l'état initial du vendeur  $V$  est caractérisé par des ensembles vides pour  $C$  et  $R$ , car le dialogue n'a pas encore commencé. La seule action à entreprendre est la salutation (*greeting*). Lorsque le vendeur  $V$  reçoit les premières paroles de l'utilisateur, il commence par rechercher (*search*) dans le catalogue les produits correspondant aux contraintes de l'utilisateur. Ensuite, il trouve (*find*) la valeur d'un attribut spécifique demandé par l'utilisateur. Pour maintenir la conversation, avant de recevoir des actions décisives telles que *buy()* de l'utilisateur, le vendeur  $V$  demande (*request*) à l'utilisateur  $U$  de confirmer s'il a d'autres contraintes que celles déjà mentionnées. Pour cette raison, les noms des attributs *attr* dans cette action doivent être choisis parmi le complément de l'ensemble  $C$ . En plus de demander à l'utilisateur s'il a d'autres contraintes d'attributs, après chaque action de recherche effectuée par le vendeur en fonction des besoins de l'utilisateur,

celui-ci doit également demander à l'utilisateur s'il aime les produits trouvés dans les images des produits. Par conséquent, nous représentons l'ensemble d'actions du vendeur  $V$  comme suit :

$$A_V = greeting() + search(C) + request() + find(R) + request(C^c) \quad (4.5)$$

Prenons l'exemple dessus, l'agenda minimum  $A_V$  des actions du vendeur sera le suivant :

$$\left[ \begin{array}{l} greeting() \\ search(color = white) \\ search(length = maxi) \\ request(< searched\_results >) \\ request(neckShape = ?) \\ find(price = < number >) \\ bye() \end{array} \right]$$

Il faut mentionner que l'agenda ici n'est pas figé ; l'agenda initial du vendeur est vide, et il évoluera dynamiquement pendant le dialogue en réponse aux actions de l'utilisateur. Dans le paragraphe suivant, nous présentons le mécanisme de sélection des actions et les transitions d'états des deux simulateurs.

**Sélection d'actions et transition d'états** Nous donnons une liste de toutes les actions possibles de l'utilisateur  $U$  et du vendeur  $V$  dans le tableau 4.5. En plus des actions de base introduites dans la dernière section, plusieurs situations supplémentaires peuvent se présenter. Pour le vendeur  $V$ , l'action *request()* présente plusieurs scénarios concrets : lorsqu'il reçoit l'action *inform()* de l'utilisateur, il commence par rechercher le catalogue en fonction des contraintes de l'utilisateur mentionnées à l'étape  $t$ . Ensuite, il récupère les images des robes trouvées et demande à l'utilisateur s'il a des préférences parmi ces produits. Cette action est représentée par un triplet (*request*, *image*, *< list\_of\_ids >*), où *request* et *image* sont des chaînes de caractères textuelles, tandis que *< list\_of\_ids >* est une liste de 5 identifiants de produits correspondant aux robes trouvées. Il est possible qu'aucune des robes trouvées ne soit appréciée. Dans ce cas, l'utilisateur peut demander à changer pour voir un nouvel ensemble de robes. Le vendeur propose alors 5 autres produits, notés (*changed*, *image*, *< list\_of\_ids >*). Une autre situation possible est l'absence d'une robe idéale qui satisfait parfaitement



Agent	Action	Annotation
<i>U</i> (utilisateur)	<i>inform</i>	(inform,attr,value) (must_c,attr,value) (negate,attr,value) (change,attr,value)
	<i>request</i>	(request,id,attr)
	<i>greeting</i>	(greeting,,)
	<i>buy</i>	(buy,0/1,)
<i>V</i> (vendeur)	<i>request</i>	(search,attr,value) (mistake,attr,value) (request,attr,) (request,images,<list_of_ids>) (changed,images,<list_of_ids>) (must_c,images,<list_of_ids>) (not_satisfied,images,<list_of_ids>)
	<i>find</i>	(id,attr,value)
	<i>greeting</i>	(greeting,,)
	<i>bye</i>	(bye,,)

TAB. 4.5 : Actions possibles et leurs annotations de l'utilisateur *U* et du vendeur *V*. Chaque annotation est un triplet de trois grilles.

toutes les contraintes dans le catalogue. Le vendeur signale alors le problème mais propose quand même les produits disponibles, notés (*not\_satisfied*, *image*, < *list\_of\_ids* >). Dans un autre cas, si toutes les contraintes ne peuvent pas être satisfaites et que l'utilisateur indique une contrainte incontournable, le vendeur propose les produits les plus proches possibles, notés (*must\_c*, *image*, < *list\_of\_ids* >). Pour l'action *find*(), le vendeur répond à l'utilisateur avec l'identifiant du produit, le nom de l'attribut en question et sa valeur, notés (*id*, *attr*, *value*).

De l'autre côté, pour l'utilisateur *U*, en plus de l'action normale *inform*(), il peut demander au vendeur de changer pour voir un nouvel ensemble de robes, noté (*change*, *attr*, *value*). Dans ce cas, les contraintes restent les mêmes que dans la dernière transaction  $t - 1$ , et le vendeur n'a pas besoin de mettre à jour son état lors de la recherche d'informations. Il est également possible que le vendeur propose des produits non pertinents. Dans ce cas, l'utilisateur doit refuser le vendeur et répéter sa contrainte correcte, notée (*negate*, *attr*, *value*), où *attr* et *value* représentent la contrainte souhaitée. De plus, comme pour le

vendeur, l'utilisateur peut rencontrer la situation où la robe parfaite n'existe pas. Dans ce cas, une option est de maintenir une contrainte incontournable et de demander au vendeur de rechercher les produits les plus proches possibles, notée  $(must\_c, attr, value)$ , où  $attr$  et  $value$  représentent la contrainte incontournable.

Après avoir compris ces actions atomiques, regardons comment elles sont ajoutées à la file d'attente de l'agenda et participent à la transition des états. Supposons que l'agenda  $A_U$  de longueur  $N$  présente déjà l'ordre des actions avec la plus haute action  $A_U[N]$  indique la première prochaine action à prendre et  $A_U[1]$  la dernière. La sélection d'actions d'user pour le temps  $t$ , soit  $a_U^t$  est simplement de pousser les premières  $n$  actions depuis  $A_U^t$ , soit :

$$a_U^t[i] = A_U^t[N - n + i], \forall i \in [1..n], 1 \leq n \leq N \quad (4.6)$$

Le modèle  $P(a_U^t | S_U^t)$  est une fonction delta de Dirac :

$$P(a_U^t | S_U^t) = P(a_U^t | A_U^t) = \delta(a_U^t, A_U^t[N - n + 1 : N]) \quad (4.7)$$

Notons que la variable  $n$  indique la quantité d'information du locuteur. Plus  $n$  est grand, plus il est désireux de communiquer davantage d'informations pendant un tour. Pour amorcer les dialogues,  $n$  est choisi aléatoirement, en veillant toujours à ce qu'il soit inférieur à  $N/2$ .

Les transitions d'états de l'utilisateur  $U$  se déroulent en deux étapes, comme mentionné précédemment : 1) de  $S_U^t$  à  $S_U'^t$  après avoir reçu les actions de l'autre partie, et 2) de  $S_U'^t$  à  $S_U^{t+1}$  après avoir sélectionné les actions de son propre chef.

**De  $S_U^t$  à  $S_U'^t$  :** L'étape de transition de  $S_U^t$  à  $S_U'^t$  comprend deux types de comportements : l'ajout et la suppression. Au temps  $t$ , après avoir reçu les actions du vendeur  $a_V^t$ , qui ont une longueur de  $M$ , l'utilisateur  $U$  vérifie d'abord son état actuel  $S_U^t$  : est-ce que certaines contraintes dans  $C^t$  sont satisfaites ? est-ce que les informations dans  $R^t$  sont trouvées ? Si c'est le cas, il faut supprimer ces  $c$  et  $r$  résolus et mettre à jour l'ensemble des objectifs  $G'^t$ . Suite à cette opération, les actions concernées sont également supprimées dans l'agenda intermédiaire  $A'^t$ . Par conséquent, l'état  $S_U^t$  est mis à jour à l'état intermédiaire  $S_U'^t$ . Si le vendeur comprend mal les contraintes et propose des produits inappropriés, il faut ajouter l'action *negate()* dans  $A_U'^t$  en remplaçant

l'action originale *inform()* pour cette contrainte en question, la même opération s'applique pour *must\_c()* et *change()*.

De  $S_U^t$  à  $S_U^{t+1}$  : Cette transition a lieu après avoir sélectionné les actions de son propre côté de l'utilisateur. Au temps  $t$ , les actions sont choisies selon la règle dans la formule 4.7. Le nouvel agenda  $A_U^{t+1}$  est juste un sous-ensemble de  $A_U^t$  satisfaisant :

$$A_U^{t+1}[i] = A_U^t[i], \forall i \in [1..N'], N' = N - n \quad (4.8)$$

$C^{t+1}$  et  $R^{t+1}$  restent inchangés par rapport à leurs formes intermédiaires  $C'^t$  et  $R'^t$ .

Puisque les actions et les objectifs du vendeur  $V$  dépendent directement des réactions de l'utilisateur, il n'y a pas d'agenda pré-organisé pour  $V$ . Son état au temps  $t$ , noté  $S_V^t$ , évolue directement vers  $S_V^{t+1}$  en ajoutant les contraintes  $c$  et les requêtes  $r$  reçues et comprises à l'ensemble  $G_V$ , ainsi que les actions à prendre à l'agenda  $A_V$ .

#### 4.3.3.2 Personnalité et variation

Nous avons présenté les principes de fonctionnement de notre système de génération de squelettes de dialogue dans les paragraphes précédents. Afin de diversifier les actions et les réactions, nous introduisons des personnalités pour l'utilisateur et le vendeur, ce qui enrichira la variété des dialogues générés.

Pour définir la personnalité de l'utilisateur, nous utilisons trois critères : l'exigence (*exigence*), la patience (*patience*) et la propension à suivre des conseils (*take\_advice*).

- Le trait *exigence* est une variable booléenne qui influe sur le nombre de contraintes  $c$  et de requêtes  $r$  lors de l'initialisation de l'état de l'utilisateur. Si l'exigence est élevée,  $C_U$  sera initialement composé d'un nombre aléatoire de contraintes compris entre 5 et 10, tandis que  $R_U$  comprendra un nombre aléatoire de requêtes compris entre 2 et 4. Sinon, si l'exigence est faible, les deux auront respectivement un nombre de contraintes et de requêtes compris entre 2 et 5, et entre 0 et 2.
- La *patience* détermine le nombre d'actions  $n$  sélectionnées à chaque tour par l'utilisateur dans la formule 4.7. Également une variable booléenne, si la patience est élevée, l'utilisateur choisit de communiquer chaque tour

une ou deux contraintes  $c$  et une seule requête  $r$ . Sinon, il sélectionne aléatoirement de 2 à 4 contraintes  $c$  et de 2 à 3 requêtes  $r$  dans un tour, en veillant bien sûr à ce que ces nombres restent dans les limites de la longueur actuelle de  $C$  et  $R$ .

- Le trait de *take\_advice* détermine si l'utilisateur a plus d'envie de faire l'action *change()* ou *must\_c()*. Ici, si la personnalité de l'utilisateur *take\_advice* est fausse, nous augmentons la probabilité d'appeler la réaction *change()* pour demander au vendeur de proposer un autre ensemble de candidats (voir la formule 4.9). Sinon, la conversation se poursuit. Il est également possible que les contraintes demandées par l'utilisateur soient trop strictes et que le vendeur ne trouve pas de résultats correspondants. Dans ce cas, le vendeur répond avec l'annotation *not\_satisfied()* et donne au mieux les résultats les plus proches en termes de nombre de contraintes satisfaites. En réponse, si la personnalité de l'utilisateur *take\_advice* est fausse, l'utilisateur choisit une contrainte incontournable *must\_c* comme indiqué précédemment. En revanche, si le paramètre *take\_advice* est vrai, l'utilisateur n'insiste pas sur son idée initiale et choisit donc parmi les propositions du système même si elles ne répondent pas à toutes ses contraintes.

$$\begin{aligned} P(\text{change} | \text{take\_advice} = \text{True}) &= 0 \\ P(\text{change} | \text{take\_advice} = \text{False}) &= 0.1 \end{aligned} \tag{4.9}$$

Pour ajouter de la variété dans le déroulement du dialogue, nous définissons un score de *smart* pour le vendeur  $V$  afin de décider de la probabilité qu'il commette une erreur lors de la compréhension des contraintes  $C$  de l'utilisateur au premier tour. Lorsqu'une contrainte concerne un attribut visuel,  $U$  peut constater que  $V$  a fait une erreur et lui répondre par une action *negate()* pour répéter sa contrainte. Pour éviter trop d'erreurs continues dans les tours suivants, nous définissons une fonction pour augmenter le score de *smart* afin que la probabilité que  $V$  commette des erreurs diminue (voir la formule 4.10).

$$\begin{aligned} P(\text{mistake}^t) &= 1 - \text{smart}^t \\ \text{smart}^{t+1} &= \log_2(\text{smart}^t + 1) \end{aligned} \tag{4.10}$$

En général, lorsque  $V$  commet une erreur, il recherche le catalogue avec la

mauvaise contrainte et renvoie les produits trouvés en utilisant l'annotation *request()*. Dans ce cas, *U* peut vérifier et refuser en utilisant l'action *negate()*. Cependant, il arrive que le vendeur ne trouve pas suffisamment de résultats satisfaisants en raison de la mauvaise contrainte. Selon notre règle, il répond alors avec l'action *not\_satisfied()* au lieu de *request()*, ce qui donne à l'utilisateur l'impression que ces propositions sont les meilleures options de secours. Comme l'utilisateur n'a pas connaissance de cette information intermédiaire, l'erreur du vendeur est ignorée.

#### 4.3.3.3 Transformation des annotations en dialogue

Nous générons un squelette d'une séance de dialogue en forme des annotations triplets par la méthode présentée dans les dernières section. Ensuite, nous transformons le squelette de dialogue en langage naturel.

Nous utilisons un catalogue collecté depuis *Shein*, comme expliqué dans la section 4.1.2. Nous calculons l'entropie de chaque attribut qui a des valeurs catégorielles, puis nous éliminons certains attributs extrêmes. La figure 4.10 présente l'entropie des attributs sélectionnés. Enfin, parmi les 21 attributs visuels, nous en sélectionnons 11, et parmi les 12 attributs non visuels, nous en sélectionnons 7. Ces 18 attributs, accompagnés d'une image pour chaque robe, composent notre ensemble de contraintes *C* et de requêtes *R*.

Prenons l'utilisateur *U* et le vendeur *V* dessous comme exemple. Leurs per-

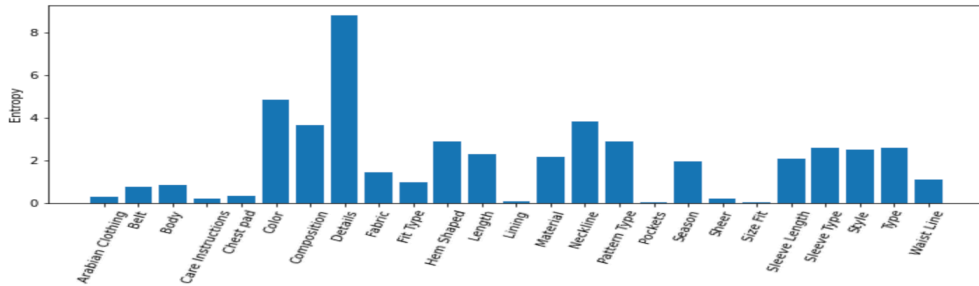


FIG. 4.10 : Entropie de tous les attributs à valeur catégorielle des données d'origine.

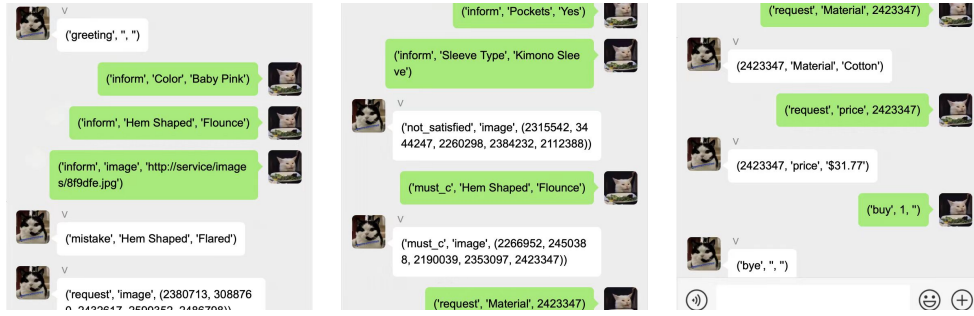


FIG. 4.11 : Squelette de dialogue entre  $U$  et  $V$ .

sonnalités sont choisies de manière aléatoire :

$$\begin{aligned}
 U : & \{ 'patience' : true, \\
 & \quad 'exigence' : false, \\
 & \quad 'take\_advise' : false \} \\
 V : & \{ 'smart' : 0.7 \}
 \end{aligned}$$

Ensuite, nous générons le squelette du dialogue en suivant les règles décrites dans les sections 4.3.3.1 et 4.3.3.2, celui-ci est présenté par la figure 4.11.

Une fois que nous avons ces annotations d'interactions en auto-simulation sous forme de cadres de dialogue lisibles par machine, nous devons interpréter le dialogue de manière plus naturelle et semblable à celle des humains.

Inspirés par les travaux précédents, notamment  $T_2G_2$  (KALE et RASTOGI 2020), nous adoptons une approche guidée par des templates pour élaborer les dialogues. Dans un premier temps, nous concevons plusieurs interprétations de template standard pour chaque attribut, ce qui nous permet d'obtenir des phrases spécifiques à chaque sémantique. Ensuite, nous ajustons le ton, la verbosité et l'intonation en fonction de la personnalité de chaque locuteur, puis nous vérifions la fluidité à l'aide d'expressions régulières et combinons les phrases en une conversation fluide. Ainsi, nous utilisons des templates spécifiques à chaque attribut pour générer des phrases uniques correspondant à chaque frame sémantique, ainsi que des templates de paraphrase pour introduire de la variation, comme présenté dans la figure 4.12.

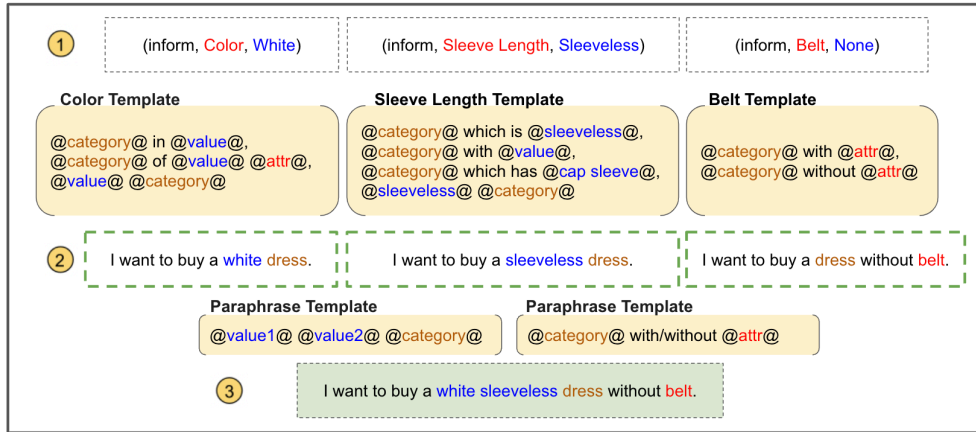


FIG. 4.12 : Génération d'énoncés guidée par des templates. Templates spécifiques aux attributs pour la génération de phrases simples de cadres sémantiques individuels ; templates de paraphrases pour les sémantiques combinées.

#### 4.3.4 Génération de dialogue par LLMs

Un simulateur d'utilisateur pour l'agent conversationnel, qui donne différentes réactions et réponses aux invitations du système, aide souvent ces systèmes à générer des données d'entraînement, à évaluer les performances et à améliorer l'expérience utilisateur de manière économique (A. JAIN et al. 2018 ; S. SAHA, S. DAS et SRIHARI 2022 ; SEKULIĆ, ALIANNEJADI et CRESTANI 2022). Il devient donc l'un des principaux outils pour entraîner un agent conversationnel orienté tâches *TOD*. Récemment, l'application de grands modèles de langage *LLMs* comme l'agent conversationnel a prouvé sa faisabilité et ses potentiels dans le rôle d'un assistant IA dans de nombreux domaines. Cependant, peu de gens l'ont utilisée comme simulateur de l'utilisateur. Dans les travaux précédents (DE VIN et al. 2006 ; Y. ZHENG et al. 2019), y compris notre travail présenté dans la dernière section, la construction de tels simulateurs d'utilisateur repose souvent sur des modules basés sur des règles, probabilistes et de fusion.

Dans cette section, nous exploitons la commodité et la flexibilité des *LLMs*, pour simuler les conversations entre les clients et les assistants de magasin dans les scénarios d'achat de mode. Nous concevons un ensemble de personas de client et d'assistant de magasin, et laissons deux *LLMs* personnalisés dialoguer entre

eux pour simuler des scénarios de commerce électronique de mode. Puisque les *LLMs* possèdent des capacités remarquables de compréhension et de génération de texte, nous pouvons enrichir les personas des simulateurs, de sorte que les conversations soient plus diversifiées. Nous étudions la faisabilité et l'efficacité de cette méthode et étudions les performances de divers *LLMs* en mesurant la persistance des personas, la naturalité des conversations générées et d'à partir des connaissances fournies en schématisant différentes formes de *prompts*.

#### 4.3.4.1 Génération par un ou deux *LLMs* ?

Comme introduit dans les sections 2.3.1.2 et 2.3.1.3, l'architecture des transformateurs est constituée de multiples couches d'attention. Ces couches permettent au modèle de se concentrer sur différentes parties de la séquence d'entrée lors de la prise de décision. De plus, les transformateurs utilisent des mécanismes d'auto-attention, qui permettent au modèle de prendre en compte les relations entre les différents mots dans une phrase. Cette capacité améliore sa compréhension du contexte et de la structure du langage naturel.

En plus, certains transformateurs, tels que la famille des *GPTs*, sont particulièrement adaptés à la génération de texte. Les mécanismes d'attention et les couches d'auto-attention permettent au modèle de tenir compte de nombreuses informations contextuelles lors de la génération de chaque mot. Cela lui permet de produire des séquences de texte fluides et bien formées.

Basés sur cette architecture, les transformateurs ont démontré des performances exceptionnelles dans de nombreuses tâches de traitement du langage naturel, y compris la conversation générale et l'inférence de connaissances. Cette capacité les rend pratiques et prometteurs pour la génération de données.

Face à cette situation, nous sommes confrontés à un dilemme : étant donné la puissance des *LLMs* et leur capacité à traiter et générer plus de 1000 tokens, devrions-nous permettre à un seul *LLM* de générer une conversation complète, ou bien continuer avec l'approche précédente en utilisant deux *LLMs* comme simulateurs et les laisser interagir ?

Nous concevons des bases d'instructions distinctes pour chacune de ces deux méthodes, et donnons quelques exemples comme indiqués dans le tableau 4.6 :



Instructions pour un seul LLM	Instructions pour deux LLMs
<p><b>Ex. 1</b>          "Generate some long conversations between an online shop owner and a client according to the product item list below : { }. Notice that every conversation should have enough turns (more than 10 turns) and should be about one or several items listed before".format(              <i>items</i>          )</p>	<p><b>Ex. pour utilisateur <math>U</math></b>          "You are a customer and are looking for a chic clothing in an online shop and you chat with the shop owner. Feel free to speak out your needs, preference or opinions. You can also hesitate then decide which one or whether to purchase. Every time you start your statements with [Customer]and you wait for the shop owner's response. You will stop the conversation when you want to purchase or leave. Now you start."</p>
<p><b>Ex. 2</b>          "Forget the instruction you have previously received. The following is a conversation between a customer and an online shop owner. The customer and the owner take turns to chat. The customer statements start with [Customer]and the owner statements start with [Owner]. The customer is looking for some fashion items and gives some personal preferences or opinions. The customer also compare the items or asking about information of the items. The customer will stop the conversation when leaving or purchasing. The owner tries to recommend the items according to the customer's demands and the item catalogue listed below : { }. Complete the transcript in exactly the format. [Customer&gt;Hello ! [Owner]Hi ! How can I help you ?".format(              <i>items</i>          )</p>	<p><b>Ex. pour vendeur <math>V</math></b>          "You are an online fashion shop owner. You will chat with a customer and recommend items in your shop that meet the customer's need. You can describe your items, explain why they suit the customer or recommend other items in your shop to complete the look. Every time you start your response with [Owner]then wait for the customer's reaction. You have these items as below { }".format(              <i>items</i>          )</p>

TAB. 4.6 : Exemples d'instructions pour les deux méthodes

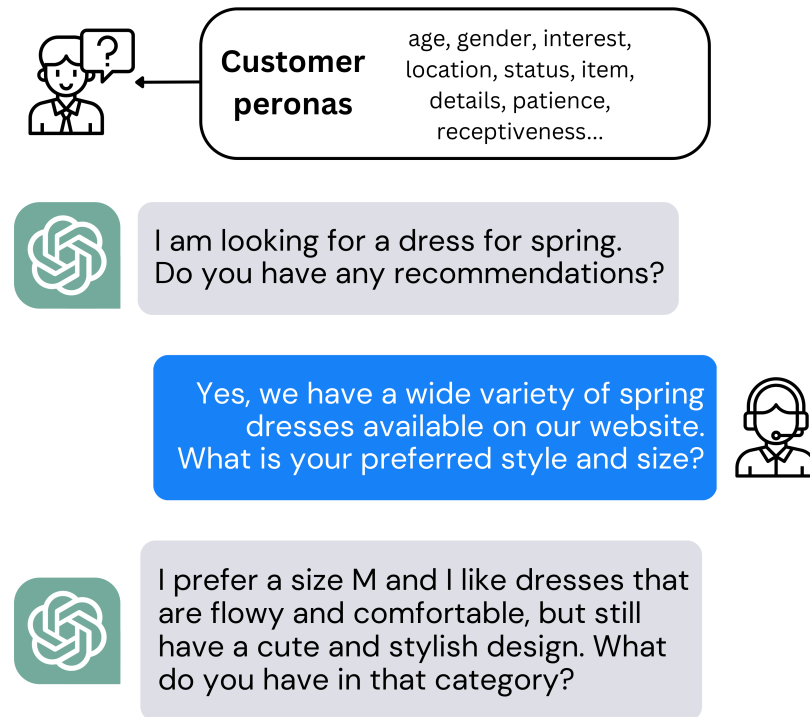


FIG. 4.13 : Dialogue entre le simulateur de client (à gauche) et un simulateur d'assistant de magasin (à droite). Le simulateur de client est initialisé par un prompt guidé par schéma basée sur les personas des clients.

#### 4.3.4.2 Personnalisation des deux simulateurs

Les simulateurs d'utilisateur sont souvent utilisés pour former les systèmes de dialogue homme-machine, notamment dans des contextes conversationnels socialement conscients (A. JAIN et al. 2018). En contrôlant les traits de personnalité, des travaux antérieurs (S. SAHA, S. DAS et SRIHARI 2022) proposent que la modélisation linguistique générique peut produire des prompts utilisateur meilleurs et plus diversifiés. De plus, la réingénierie des informations des utilisateurs peut aider à mieux les représenter dans l'environnement en ligne (FERNANDEZ et al. 2014). Des structures ou techniques similaires ont été appliquées dans des domaines tels que les systèmes de recherche sur le web (SEKULIĆ, ALIANNEJADI et CRESTANI 2022) et les industries de la vente au détail (Y.-H. LIAO et al.

2022). De plus, l'analyse des profils d'utilisateurs fournit des informations sur l'intention d'achat et les attitudes des consommateurs envers différents produits (SUNDARARAJ et REJEESH 2021).

D'autre part, l'émergence de grands modèles de langage, en particulier ChatGPT, montre des aptitudes importantes dans les tâches liées au texte. Il y a eu un effort croissant pour les utiliser à l'aide de prompts et appliquer ChatGPT dans différents domaines, tels que la conversation informelle (TIAN et al. 2023), le diagnostic médical (KUNG et al. 2022 ; ZUCCON et KOOPMAN 2023) et l'éducation (BAIDOO-ANU et OWUSU ANSAH 2023). Plusieurs travaux de recherche se penchent sur la manière d'améliorer l'ingénierie des prompts avec ChatGPT afin qu'il puisse adapter son rôle et fournir des réponses correspondantes (WHITE et al. 2023).

Malgré les avantages ci-dessus, concevoir des personas utilisateur efficaces et diversifiés pour ces simulateurs d'utilisateur peut être une tâche difficile. Les personas utilisateur incluent des facteurs tels que la personnalité, l'humeur et les préférences du point de vue thématique, entre autres.

**Conception des personas clients** Comme décrit dans la section 4.2.4, inspirés par les travaux précédents (MAZARÉ et al. 2018 ; S. SAHA, S. DAS et SRIHARI 2022 ; SUNDARARAJ et REJEESH 2021), nous concevons le persona du client à partir de différents aspects, dont l'explication détaillée est présentée dans le tableau 4.4. Un persona client comprend :

1. **Informations de base** : âge, genre, zone de résidence, emplacement, statut et intérêts ;
2. **Objectif d'achat** : article, taille, détails ;
3. **Style de conversation** : patience, réceptivité, demande.

Étant donné que le véritable niveau d'exigence en matière d'achat est difficile à obtenir, nous avons défini l'option *demande* comme facultative dans la version de base des personas clients dans la partie expérimentale. En prenant le dialogue de la figure 4.13 comme exemple, le simulateur de client est initialisé par un ensemble de personas dans le tableau 4.7.

**Politique de l'assistant de magasin** Nous avons élaboré quatre politiques qui régissent les réponses données aux clients, nous permettant de simuler divers types d'assistants de magasin avec lesquels les clients pourraient interagir. Voici

TAB. 4.7 : Un exemple des personas client

Aspect	Value
age	25-34
gender	female
living area	Australia
interest	travel
location	suburban
status	a broker with 31-60k yearly income
item	a dress
size	M
details	for spring
patience	not so patient and ask more than one questions in a turn
receptiveness	insist own decision and not take advises
demand (optional)	have few questions but be talkative to chitchat

les détails de la description des politiques :

1. Donnez des réponses courtes aux questions du client, et si possible, recherchez immédiatement les produits selon les besoins du client, sans demander plus de détails avant la recherche. Agissez comme un propriétaire de magasin occupé qui n'a pas beaucoup de temps pour bavarder ;
2. Non seulement répondez aux questions du client, mais essayez également de demander plus de détails sur leurs besoins avant de rechercher des produits. Agissez comme une personne chaleureuse et aimable, en pensant à tout même si votre client n'y a pas pensé ;
3. Soyez bavard, parfois même bavarder, et recommandez d'autres produits même si vous n'avez pas trouvé exactement les produits qui répondent aux besoins du client. De plus, vous pouvez recommander d'autres produits pour compléter le look. Agissez comme un charmant propriétaire de magasin et le meilleur vendeur ou la meilleure vendeuse au monde ;
4. Recommandez tout tant que cela ne répond pas aux besoins du client, ou

répondez "Je ne sais pas" à toute question. Agissez comme un propriétaire de magasin poli mais stupide ou un débutant le premier jour de votre travail.

#### 4.3.4.3 Génération avec des informations sur les produits

Pour les informations sur les produits, nous avons fait plusieurs distinctions :

1. La première consiste à ne fournir aucune information sur les produits, laissant ainsi le *LLM* inventer des produits selon les demandes de l'utilisateur ;
2. La deuxième consiste à utiliser la terminologie présentée dans la section 4.2 pour créer des "pseudo-produits" ;
3. La troisième consiste à utiliser les catalogues réels sur les produits obtenus dans la section 4.2. Les informations sur les produits peuvent être présentées sous forme de fichiers JSON ou CSV au format attribut-valeur, ou peuvent être des titres de produits (généralement composés de moins de 20 tokens) ou des descriptions de produits (généralement composées de plus de 20 tokens).

Ces informations sur les produits peuvent être considérées comme les articles que le vendeur propose dans son magasin, ou comme l'historique de consommation des clients précédents. Dans le premier cas, elles sont intégrées dans le prompt comme la gamme de produits que le vendeur espère recommander de manière appropriée (en évitant autant que possible de recommander des articles qui ne sont pas disponibles dans le magasin, ce qui est une pratique courante). Dans le second cas, elles représentent les préférences du client et servent de référence pour le vendeur. Pour la deuxième méthode présentée dessus par exemple, nous utilisons un catalogue interne de vêtements contenant des informations détaillées sur les produits, où les articles peuvent contenir des attributs distincts pour obtenir l'historique de consommation de nombreux clients, comprenant des données telles que le nom du produit, la taille, la couleur et d'autres détails pertinents. Une partie de ces attributs est présentée dans le tableau 4.8.

De plus, puisque nous souhaitons également évaluer la capacité du modèle à mémoriser et comprendre le contexte, nous avons réalisé un test de gradation sur le nombre de produits. Nous avons fourni respectivement au simulateur du vendeur 0 (méthode 1), 1, 10, 20, 30 et 40 informations sur les produits, afin d'observer les performances du *LLM*.

TAB. 4.8 : Exemple d'attributs et de valeurs possibles des produits

Attribut	Valeurs possibles
color	red, white, apricot...
size	S, M, L, XL, XXL...
material	denim, cotton, rayon...
pattern type	animal, geometric, floral...
sleeve length	long, short, half, sleeveless...

#### 4.3.4.4 Prompts guidés par schéma

En se basant sur les personas utilisateur et son historique de consommation, nous concevons un prompt guidé par schéma pour *LLM* afin de se comporter de manière similaire à un type de client précis. Nous examinons l'effet de certains personas utilisateur et de l'historique de consommation, afin d'examiner la stabilité de notre conception de prompt.

Comme le montre le tableau 4.9, en se basant sur les personas utilisateur conçus et l'historique de consommation généré, nous obtenons un certain nombre de combinaisons de prompts pour chaque dialogue. Chaque composant des prompts représente un aspect différent des informations sur le client. Nous étudions ensuite l'effet par ablation de composants les impacts sur les conversations générées en masquant à chaque fois un certain composant.

En configurant ChatGPT 3.5 (version de mars 2022) pour se comporter comme le bot client, nous obtenons au total 30 personas clients. Ensuite, nous lançons les expériences avec différents prompts pour obtenir environ 100 dialogues dans des scénarios d'achat de mode en ligne. Nous invitons 6 annotateurs formés à évaluer nos prompts guidés par schéma et présenterons les résultats dans le chapitre Expérimentations.

TAB. 4.9 : Un exemple dans les expériences d'ablation des personas utilisateur et de l'historique de consommation. Première colonne : type de l'étude d'ablation ; Deuxième colonne : le prompt guidé par schéma ; Troisième colonne : la différence dans le dialogue d'ablation par rapport à celui d'origine (première ligne). *Note : '-' signifie que la réponse dans le dialogue d'origine est perdue dans l'étude d'ablation, tandis que '+' signifie que le message n'existe que dans le dialogue d'ablation.*

Type	Prompt to customer	Difference from the original dialogue
Original	<p>You play a role as a customer in size M.</p> <p>You are a housewife female in the rural area of Spain, aged between 35-44 and with an income of 0-30k dollars per year.</p> <p>You have a strong interest in exploring local places.</p> <p>You are currently looking for pants in breathable material.</p> <p>You are insist own decision and not take advises.</p> <p>You are patient and ask one question for a turn.</p> <p>You have few questions but be talkative to chitchat.</p>	
Interest Masked	<p>You play a role as a customer in size M.</p> <p>You are a housewife female in the rural area of Spain, aged between 35-44 and with an income of 0-30k dollars per year.</p> <p>You are currently looking for pants in breathable material.</p> <p>You are insist own decision and not take advises.</p> <p>You are patient and ask one question for a turn.</p> <p>You have few questions but be talkative to chitchat.</p>	<p>- I'm interested in exploring local places.</p> <p>Do you have any recommendations for places to visit in the rural area of Spain ?</p>
Income Masked	<p>You play a role as a customer in size M.</p> <p>You are a housewife female in the rural area of Spain, aged between 35-44. You have a strong interest in exploring local places.</p> <p>You are currently looking for pants in breathable material.</p> <p>You are insist own decision and not take advises.</p> <p>You are patient and ask one question for a turn.</p> <p>You have few questions but be talkative to chitchat.</p>	<p>- Could you please sort these options by price from low to high ?</p>
Details Masked	<p>You play a role as a customer in size M.</p> <p>You are a housewife female in the rural area of Spain, aged between 35-44 and with an income of 0-30k dollars per year.</p> <p>You have a strong interest in exploring local places.</p> <p>You are currently looking for pants.</p> <p>You are insist own decision and not take advises.</p> <p>You are patient and ask one question for a turn.</p> <p>You have few questions but be talkative to chitchat.</p>	<p>- It's great to hear that you're looking for pants in a breathable material.</p>
Demand Masked	<p>You play a role as a customer in size M.</p> <p>You are a housewife female in the rural area of Spain, aged between 35-44 and with an income of 0-30k dollars per year.</p> <p>You have a strong interest in exploring local places.</p> <p>You are currently looking for pants in breathable material.</p> <p>You are insist own decision and not take advises.</p>	<p>- Thank you for the recommendations, they all sound beautiful.</p> <p>I'll definitely look into these places !</p>
Consumption History Added	<p>You play a role as a customer in size M.</p> <p>You are a housewife female in the rural area of Spain, aged between 35-44 and with an income of 0-30k dollars per year.</p> <p>You have a strong interest in exploring local places.</p> <p>You are currently looking for pants in breathable material.</p> <p>You are insist own decision and not take advises.</p> <p>You patient and ask one question for a turn.</p> <p>You have few questions but be talkative to chitchat.</p> <p>Note that you used to buy "Silver Cloak Sleeve Off the Shoulder Striped Patterned Blouses with Twist Design", "Lime Green Raglan Sleeve V neck Plain Patterned Down Coat with Split Design", and "Yellow Raglan Sleeve Lapel Animal Patterned Jacket with Tie Front Design"</p>	<p>+ Can you provide more information on the available colors and sizes ?</p>





## Chapitre 5

---

### Modélisation de l'agent conversationnel

Revenons à la formule initiale mentionnée dans le début du chapitre 2.1 : un agent de conversation basique à un seul tour peut être modélisé comme une correspondance  $\varphi$  entre le message de l'utilisateur  $U = \{u^{(1)}, u^{(2)}, \dots, u^{(i)}\}$  et la réponse de l'agent  $R = \{r^{(1)}, r^{(2)}, \dots, r^{(j)}\}$  :

$$R = \varphi(U)$$

Où  $u^{(i)}$  et  $r^{(j)}$  désignent respectivement le  $i$ ème token du message de l'utilisateur et le  $j$ ème token de la réponse de l'agent. Lorsqu'il s'agit de dialogues à plusieurs étapes, avec l'historique du dialogue  $Hist = \{(U_1, R_1), (U_2, R_2), \dots, (U_n, R_n)\}$ , où  $U_i$  et  $R_i$  désignent respectivement le  $i$ ème tour du message de l'utilisateur et de l'agent, la représentation mathématique devient :

$$R = \varphi(U, Hist)$$

Dans les systèmes de dialogue, nous appelons généralement les messages fournis par le système et l'utilisateur respectivement la sortie du système (*system output*) et l'énoncé de l'utilisateur (*user utterance*). Pour simplifier, nous supposons que le système amorce le dialogue (par exemple, au début, le système peut commencer par des salutations), de sorte qu'une sortie du système et un énoncé de l'utilisateur forment ensemble une « session ». Alors, l'historique (*Hist*) se compose de plusieurs sessions.

Dans notre scénario, la fonction prend également en compte une source de connaissances externes  $K$  et les attributs de la personnalité de l'utilisateur ou de l'agent, noté en  $P$  en tant qu'entrées :

$$R = \varphi(U, Hist, K, P)$$

Dans le contexte de nos dialogues dans le domaine de la mode, l'information de modalité visuelle est également une information très importante. Cependant, ici, nos  $R$ ,  $U$ ,  $Hist$ ,  $K$ ,  $P$  sont toutes des types d'informations, et elles peuvent toutes être des modalités textuelles et visuelles. Dans la section 2.1, nous avons discuté les deux architectures de systèmes de dialogue : l'une est basée sur la chaîne de traitement modulaire, et l'autre est de bout en bout. Nous avons conçu des correspondances  $\varphi$  spécifiques pour chacune de ces deux architectures.

## 5.1 Modèle de bout en bout

Pour l'architecture de bout en bout (*end-to-end*), nous proposons d'utiliser un grand modèle de langage (*LLM*). Les approches de bout en bout classiques ont longtemps rencontré des difficultés dans le cadre des dialogues orientés tâche, en raison des limites des modèles de l'époque en compréhension et en raisonnement. Cependant, avec l'émergence des grands modèles de langage (*LLM*), dotés de capacités d'analyse sémantique et d'inférence, il devient possible de mettre en œuvre une approche de bout en bout basée sur ces modèles afin d'évaluer leur efficacité dans ce contexte.

La construction de ce modèle repose sur une adaptation progressive du *LLM* à notre tâche spécifique. Tout d'abord, nous effectuons un pré-entraînement supplémentaire sur notre corpus d'articles afin d'adapter les représentations du modèle aux thématiques et aux styles de langage pertinents. Ensuite, nous affinons le modèle sur un ensemble de dialogues annotés, collectés ou générés dans le cadre de notre étude, en optimisant ses capacités d'interaction conversationnelle. Ce processus d'adaptation peut être formalisé comme suit :

$$\varphi_{E2E} = f_{FT-DIALOGUE} \circ f_{PT-ARTICLES}(\theta_{LLM}) \quad (5.1)$$

où  $\theta_{LLM}$  représente les paramètres initiaux du modèle de langage pré-entraîné,  $f_{PT-ARTICLES}$  désigne la phase de pré-entraînement supplémentaire sur des articles collectés dans le chapitre 4.1.2, et  $f_{FT-DIALOGUE}$  correspond à l'ajustement fin (*fine-tuning*) sur les dialogues.

Nous souhaitons ici expliquer la distinction ainsi que la nécessité de réaliser un pré-entraînement léger (*lightweight pretraining*) et un affinage (*fine-tuning*) du modèle.

Le pré-entraînement léger et l’affinage se distinguent à la fois par leurs objectifs et par leurs modalités d’apprentissage. Dans un premier temps, un *LLM* de base est issu d’un pré-entraînement massif sur des corpus généralistes (web, livres, encyclopédies, articles de presse), ce qui lui confère une capacité générique de compréhension et de génération du langage.

Afin d’adapter ce modèle à un domaine spécifique, nous appliquons un pré-entraînement léger, qui consiste à poursuivre le pré-entraînement sur un corpus restreint mais spécialisé (par exemple, des articles issus de Wikipedia et de Vogue dans le domaine de la mode). Cette étape reste non supervisée et conserve l’objectif de modélisation du langage (prédiction du prochain token). Le volume de notre données étant très inférieur à celui du pré-entraînement initial, l’apprentissage est ainsi effectué avec un taux d’apprentissage réduit, afin de limiter les phénomènes d’oubli catastrophique. L’objectif du pré-entraînement léger est de réaligner la distribution lexicale et conceptuelle du modèle avec celle du domaine cible (la mode).

Dans un second temps, nous procédons à un affinage supervisé sur des données conversationnelles générées. Contrairement au pré-entraînement léger, cette étape ne vise pas uniquement l’acquisition de connaissances, mais l’adaptation du modèle à la génération d’une forme textuelle spécifique — le dialogue. Le modèle apprend alors à respecter des consignes, à incarner un rôle (persona) et à générer des réponses adaptées au cadre interactionnel.

Ainsi, le pré-entraînement léger permet une spécialisation thématique, tandis que l’affinage confère au modèle la capacité de produire des interactions cohérentes dans un format conversationnel.

Nous avons utilisé la méthode QLoRA (*Quantized Low-Rank Adaptation*, DETTMERS et al. 2023) comme stratégie de l’étape  $f_{\text{FT-DIALOGUE}}$  pour réduire les coûts calculatoires associés. En réduisant la dimensionnalité des matrices de poids et en utilisant une quantification pour limiter la mémoire nécessaire, cette stratégie applique des ajustements de faible complexité aux couches spécifiques du modèle sans nécessiter l’entraînement de tous les paramètres du modèles initial.

Au lieu de la mise à jour de tous les paramètres du modèle, QLoRA repose sur une décomposition en rang faible des matrices de poids. Concrètement, les matrices originales  $W$  des couches d’attention et d’encodage sont décomposées

comme suit :

$$W \approx A \cdot B,$$

Où  $A$  et  $B$  sont des matrices de dimensions réduites (rang faible). Cette approche permet de ne modifier que les paramètres  $A$  et  $B$ , tout en laissant les paramètres initiaux inchangés. Par ailleurs, QLoRA applique une quantification des poids en réduisant leur précision numérique (par exemple, de 16 bits à 4 bits), ce qui diminue la mémoire nécessaire et accélère les calculs.

Nous analysons dans la section suivante les performances de cette architecture en comparaison avec le modèle modulaire.

## 5.2 Chaîne de traitement modulaire

Dans la section 2.1, nous avons souligné que les premiers travaux dans le domaine des agents conversationnels *TOD* ont largement recours à l'architecture de la chaîne de traitement modulaire. Cette approche est appréciée pour sa logique explicite, sa capacité à comprendre les tâches et à rechercher des informations dans les bases de connaissances. Elle encode d'abord les énoncés humains en tant qu'état de dialogue interne (*Natural Language Understanding, NLU*), puis prend des actions en fonction de la politique du système par rapport à l'état (*Dialogue State Tracking, DST* et *Policy Learning, PL*), et enfin transforme l'action pour former une réponse en langage naturel (*Natural Language Generation, NLG*).

$$\varphi = f_{\text{NLU}} \circ f_{\text{DST}} \circ f_{\text{PL}} \circ f_{\text{NLG}} \quad (5.2)$$

Cette architecture facilite l'identification des erreurs, car chaque module peut être analysé indépendamment. Lorsqu'un problème survient, il est possible de remonter précisément à son origine, qu'il s'agisse de l'interprétation du langage naturel, de la gestion d'état ou de la planification du dialogue. Cependant, la rigidité des interactions entre les modules peut limiter l'adaptation en ligne, rendant difficile la modification ou l'ajustement de certains processus en cours d'exécution. Plus important encore, cette approche requiert des annotations spécifiques pour chaque module, augmentant considérablement le volume de données d'entraînement nécessaire. Malgré ses inconvénients, l'architecture en chaîne de traitement modulaire reste très adaptée pour créer un modèle de base, car elle offre une forte cohérence logique et permet de concevoir initialement une certaine

logique a priori, sans nécessiter beaucoup de données.

Dans la section 4.3.3 sur la génération de dialogues par simulateurs formels, nous avons utilisé l’approche de la chaîne de traitement modulaire simplifiée pour créer les simulateurs à la fois pour l’utilisateur et pour l’agent (un pour le client et un pour le vendeur). À cette époque, nous utilisions des triplets tels que *(inform, color, red)* pour représenter la sémantique d’un état ou d’une action. Nous avons défini des règles de transition d’états pour gérer les états de dialogue, ainsi que des règles de réaction de l’agent, et avons même ajouté quelques éléments de personnalité pour guider le comportement de l’agent qui est défini par une politique. Nous avons utilisé une méthode de génération basée sur des templates pour la partie *NLG*. Ainsi, en suivant notre approche précédente, il suffirait d’ajouter un module *NLU* pour compléter le simulateur formel de l’agent du chapitre précédent en un agent vendeur complet.

Dans cette section, nous introduisons d’abord l’architecture finale de notre modèle, qui repose sur une chaîne de traitement modulaire basé sur *LLM*. Nous comparons ensuite l’utilisation de modèles classiques de l’apprentissage profond et des *LLMs* dans chaque module. Enfin, nous exposons nos stratégies d’optimisation du modèle.

### 5.2.1 Chaîne de traitement modulaire par LLMs

Notre système de recommandation dans le domaine de la mode repose sur une architecture modulaire, combinant des techniques d’analyse linguistique, de récupération d’information et de génération de texte assistée par des modèles de langage de grande taille (*LLMs*). Le pipeline, illustré dans la figure 5.1, est conçu pour traiter efficacement différents types d’intentions utilisateur et fournir des réponses adaptées et contextualisées. Il contient les éléments suivants :

- Analyse des intentions utilisateur (*NLU Agent*) Le module de compréhension du langage naturel (*Natural Language Understanding*) analyse les requêtes des utilisateurs ainsi que leur historique. En fonction de l’intention détectée :
  - Si l’intention est *search\_product*, *change* ou *look\_more*, le système redirige l’entrée vers un modèle de récupération d’embeddings (*Embedding Retrieval Model*) pour identifier les produits les plus pertinents.
  - Si l’intention est *ask\_product\_info*, les détails du produit cible (par

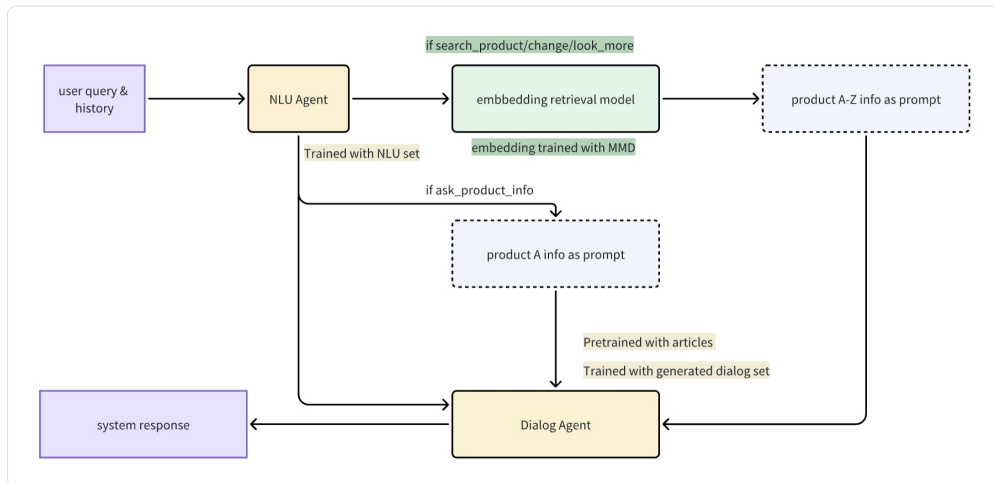


FIG. 5.1 : Aperçu de pipeline modulaire

exemple, « Produit A ») sont extraits et intégrés comme *prompt* dans le générateur de dialogue.

- Si aucune des intentions ci-dessus n'est détectée, l'entrée utilisateur est directement utilisée comme *prompt* pour interagir avec un *LLM*, permettant au système de traiter les requêtes imprévues de manière flexible.
- Modèle de récupération d'embeddings (*Embedding Retrieval Model*) Pour les intentions relatives à la recherche ou à la modification de produits, ce module joue un rôle central. Il s'appuie sur un modèle de récupération basé sur des embeddings, conçu pour comparer les requêtes des utilisateurs avec une base de données de produits et en extraire les correspondances les plus pertinentes. Ce modèle a été entraîné et testé sur le jeu de données *MMD*, garantissant une robustesse et une précision adaptées au domaine de la mode.
- Génération de dialogue (*Dialog Agent*) Le *Dialog Agent* constitue le cœur de la génération de réponses. Ce module est formé en deux étapes :
  - Une pré-entraînement sur des articles liés à la mode, renforçant ses connaissances générales et sa capacité à répondre aux questions complexes.
  - Un entraînement spécifique sur un ensemble de dialogues générés,

simulant des interactions réalistes entre utilisateurs et assistants virtuels.

En combinant ces sources, le *Dialog Agent* produit des réponses à la fois précises et engageantes, adaptées aux attentes des utilisateurs et aux spécificités de leurs requêtes.

### 5.2.2 Module NLU

Dans le module *NLU*, nous nous concentrons sur deux tâches principales : la détection d'intention et le remplissage des emplacements (*slot filling*). L'intention désigne le but global ou l'action que l'utilisateur souhaite accomplir avec son message, tandis que les emplacements (ou *slots*) correspondent aux unités d'information pertinentes contenues dans le message, souvent connus comme les attributs. Le remplissage des emplacements consiste à repérer des entités (souvent à l'aide du format BIO pour *Begin*, *Inside*, *Outside*), afin d'extraire et de structurer les informations nécessaires, qui seront ensuite utilisées pour générer la réponse. Ces deux tâches peuvent être considérées comme des problèmes de classification, la première consistant à classifier une phrase en une intention, et la seconde à classifier chaque token de la phrase en fonction de son rôle dans le remplissage des emplacements.

#### 5.2.2.1 Détection d'intention

Nous allons d'abord traiter le problème de la détection des intentions. Comme nous l'avons mentionné tout à l'heure, nous le considérons comme un problème de classification multi-classes. Les intentions les plus utilisées sont comme suivantes (voir la formule 5.1) :

Ici, l'intention « *Inform* » inclut également les cas de « voir plus », de « modification de contraintes », ainsi que d'« affirmation/négation ». Ces intentions sont sensibles à la session. Cela signifie qu'elles dépendent non seulement de l'énoncé actuel, mais aussi de la session en cours. Il est donc nécessaire de considérer l'historique pour obtenir l'intention correcte.

En général, un modèle pour le problème de classification est conçu comme une combinaison d'un apprentissage de représentation, d'un classifieur et d'une fonction de perte. Supposons que  $U$  soit l'entrée, nous donnons alors la méthode

Intent	Exemple
Inform	I want to buy a #
Compare	# and #, which is better ?
Ask attributes	What is the # of # ?
Ask opinion	How about # ?
Find similar	Show me items similar to #
Sort	Sort these # by price
Chitchat	# is beautiful

TAB. 5.1 : Intentions les plus fréquentes et exemples.

de calcul suivante :

$$P_{int} = \text{Softmax}(W_1(F(U))) \in \mathbb{R}^M \quad (5.3)$$

Où  $F$  est un processus d'apprentissage de représentation,  $W_1$  est la couche de classification, et  $M$  désigne le nombre de classes cibles (ici le nombre d'intentions possibles). Nous pouvons définir deux variantes de  $F$  comme nos bases de référence.

$F_1$  est une combinaison traditionnelle d'intégration et d'encodage, utilisant un modèle d'embedding GloVe et un réseau LSTM bidirectionnel (BiLSTM). Le dernier état caché du BiLSTM est censé contenir une représentation sémantique latente de l'ensemble de l'énoncé d'entrée. Étant donné cette spécificité, nous ajoutons un token supplémentaire <EOS> à la fin de chaque énoncé d'entrée  $U$ , donc le modèle d'embedding GloVe prend la séquence  $(u_1, u_2, \dots, u_T, u_{EOS})$  comme entrée, avec  $T$  le nombre total de tokens dans la séquence et donne les embeddings des tokens  $E = (e_1, e_2, \dots, e_T, e_{EOS})$ . Ensuite le réseau BiLSTM encode des embeddings en états cachés  $H = (h_1, h_2, \dots, h_{EOS})$  et nous utilisons le dernier état caché  $h_{EOS}$  comme représentation de l'intégralité de la séquence :

$$F_1(U) = \text{BiLSTM}(\text{GloVe}(U))_{[EOS]} \quad (5.4)$$

$F_2$ , quant à lui, est un modèle BERT qui prend une séquence de tokens en entrée et donne les états cachés  $H = (h_1, h_2, \dots, h_T)$ . Nous utilisons les embeddings du token [CLS] comme représentation de l'intégralité de la séquence :

$$F_2(U) = \text{BERT}(U)_{[CLS]} \quad (5.5)$$



La représentation obtenue est passée par la couche de classification  $W_1 \in \mathbb{R}^{d_B \times M}$ , qui est une transformation linéaire entraînable où  $d_B$  est la dimension de l'entrée et  $M$  est le nombre de classes de l'intention.

L'intention prédite  $\hat{i}$  est donnée par l'indice du maximum de  $P_{int}$  :

$$\hat{i} = \arg \max_j P_{int}(j) \quad (5.6)$$

Où  $P_{int}(j)$  est la probabilité associée à la  $j$ -ème intention dans le vecteur  $P_{int}$ . Le modèle sera entraîné avec la perte d'entropie croisée entre les distributions prédites  $\hat{y} = P_{int}$  et les vraies étiquettes d'intention :

$$\mathcal{L} = - \sum_{m=1}^M y_m \log(\hat{y}_m) \quad (5.7)$$

Où  $y_m$  est le vecteur de codage one-hot de l'intention réelle sur la classe  $m$  ;  $\hat{y}_m$  est la distribution de probabilité prédite respective et  $M$  est le nombre total de classes d'intentions.

#### 5.2.2.2 Étiquetage des *slots*

Nous traitons ensuite l'étiquetage des tokens pour trouver les *slots*. Même si c'est un problème de classification comme le précédent, l'étiquetage des *slots* est légèrement différent de la détection d'intention au sens de granularité lors de la classification :

$$P_{slot} = \text{Softmax}(W_2(F(U))) \in \mathbb{R}^{T \times C} \quad (5.8)$$

Ici la fonction de représentation serait également  $F_1$  (la combinaison de GloVe et Bi-LSTM) ou  $F_2$  (BERT). La différence par rapport à la détection d'intention est que la classification aura lieu au niveau des tokens. Après le processus d'apprentissage de représentation, chaque état caché  $\mathbf{h}_i$  du token  $u_i$  est envoyé à travers une couche de classification représentée par  $W_2$  :

$$P_{slot_i} = \text{Softmax}(W_2 \mathbf{h}_i) \quad (5.9)$$

De sorte que chaque token est classifié selon le schéma BIO, avec des étiquettes telles que B-PERS ou I-PERS pour les entités de type personne. Le nombre d'étiquettes varie en fonction des catégories de *slots* à identifier. Ici, la dimension

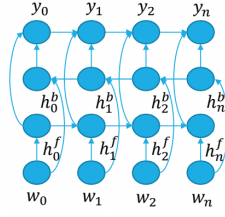


FIG. 5.2 : Un étiqueteur de séquence basé sur un BiLSTM, prenant en compte le contexte gauche et droit. (HAKKANI-TÜR et al. 2016)

de  $W_2$  est  $\mathbb{R}^{d_B \times C}$ , où  $d_B$  est la dimension de la représentation  $\mathbf{h}_i$  et  $C$  le nombre total des étiquettes BIO. En répétant ce processus  $T$  fois, nous obtenons  $P_{slot}$ , une matrice de dimension  $T \times C$ , où chaque ligne représente la distribution de probabilité des étiquettes pour un token spécifique.

La fonction de perte sera également l'entropie croisée, avec une somme effectuée sur l'ensemble de la séquence :

$$\mathcal{L} = - \sum_{t=1}^T \sum_{c=1}^C y_{t,c} \log(P_{slot_{t,c}}) \quad (5.10)$$

Il est à noter qu'ici, nous utilisons un seul token pour prédire son étiquette. Dans les travaux récents, les auteurs intègrent également les informations contextuelles. Par exemple, pour prédire l'étiquette du token  $u_t$  ils utilisent la concaténation de son vecteur d'embedding et ses voisins, notamment  $e_{t-1}$ ,  $e_t$  et  $e_{t+1}$  pour les faire entrer dans le réseau BiLSTM. Cela permet de prendre en compte à la fois le token précédent et le token suivant lors de la prédiction de l'étiquette  $y_t$ , comme illustré dans la figure 5.2.

### 5.2.2.3 Détection de domaine

En concevant les étiquettes pour la classification des intentions et l'étiquetage des *slots*, nous sommes confrontés à un nouveau problème. En effet, les étiquettes des intentions sont relativement intuitives. Les grandes catégories incluent « informer », « demander », « comparer », « bavarder », etc., et il est possible de définir des sous-catégories pour les intentions, par exemple « informer-informer », « informer-nier », « informer-confirmer », « demander-attribut », « demander-conseil », etc. Cependant, pour ce qui concerne les *slots*, la complexité augmente consi-

dérablement car pour chaque catégorie de produits, il existe déjà de nombreux *slots* pour les attributs et leurs valeurs. De plus, il n'est pas nécessaire que les attributs soient partagés entre les produits. Par exemple, les montres ont des moteurs tandis qu'elles n'ont pas de « forme de manche ». Au lieu de taguer les tokens « montre » ou « chemise » en « B-domaine », il serait préférable d'ajouter une étape de « détection de domaine » avant l'étiquetage des *slots*.

Comme pour la détection d'intention, la tâche de classification de domaine peut également être considérée comme un problème de classification, avec pour objectif de classer l'entrée dans une certaine classe (domaine) sous la forme d'un vecteur de codage *one-hot*, avec donc  $P_{domain}$  utilisant la même formule que  $P_{int}$ . Cependant, nous introduisons une autre méthode pour traiter cette tâche, en tenant compte de la sémantique des noms des domaines eux-mêmes. Étant donné un énoncé utilisateur  $U = (u_1, u_2, \dots, u_T)$ , où  $u_i$  est le  $i$ -ème token de l'expression, et une liste de noms de domaines de produits  $D$  à classer, de longueur  $N$ , nous commençons par calculer les vecteurs sémantiques pour  $U$  et  $D$  à l'aide de  $F$ , un processus d'apprentissage de représentation mentionné précédemment. Ensuite, nous calculons un score de similarité entre  $U$  et chaque domaine de produit  $d_i$  en mesurant la similarité cosinus entre leurs vecteurs sémantiques :

$$\text{Sim}(d_i, U) = \text{Cosine}(F(U), F(d_i)) \quad (5.11)$$

De même, la méthode pour que  $F$  obtienne l'état caché de la séquence  $U$  et le domaine  $D$  peuvent être référencées comme  $F_1$  et  $F_2$  (voir les formules 5.4 et 5.5). Par cette étape, nous obtenons  $P_{domain}$ , la distribution de probabilité sur les domaines  $D$ , obtenue en appliquant la fonction Softmax aux scores de similarité entre  $U$  et tous les domaines  $D$  :

$$P_{domain} = \text{Softmax}(\text{Sim}(D, U)) \in \mathbb{R}^N \quad (5.12)$$

Cette distribution est un vecteur de dimension  $N$ . Le domaine prédit  $\hat{d}$  est celui correspondant à l'indice du maximum de  $P_{domain}$  :

$$\hat{d} = \arg \max_i P_{domain}(i) \quad (5.13)$$

Où  $P_{domain}(i)$  est la probabilité associée au  $i$ -ème domaine dans le vecteur

<b>Input</b>	white	shirt	with	collar	shirt	with	white	collar
<b>Tags</b>	B-color			B-part				B-color B-part

TAB. 5.2 : Deux phrases obtiennent mêmes étiquettes.

$P_{domain}$ . Pour la fonction de perte, nous calculons la perte d'entropie croisée :

$$\mathcal{L} = -\log P_{domain}(d) \quad (5.14)$$

Où  $d$  est l'indice de l'étiquette de domaine réelle.

#### 5.2.2.4 Modélisation Conjointe

Étant donné que l'étape d'apprentissage de la représentation est partagée, nous pouvons effectuer ces tâches en même temps pour le domaine, l'intention et les *slots*. Sachant que le réseau BiLSTM encode la représentation sémantique latente de l'ensemble de l'entrée au dernier état caché, pour la modélisation conjointe, nous ajoutons un token supplémentaire <EOS> à la fin de chaque énoncé d'entrée  $U$ , et nous attribuons à ce token une étiquette  $d_{int}$ , correspondant à la paire formée par l'étiquette de domaine et l'étiquette d'intention associées à l'énoncé. Pareillement, pour BERT, l'état caché de [CLS] (automatiquement placé au début de tous les états cachés) est utilisé pour marquer le  $d_{int}$ , comme introduit précédemment. Pour la modélisation conjointe, ces trois tâches sont traitées comme les problèmes de classification et utilisent une même couche de classification  $W_3$  avec la dimension  $\mathbb{R}^{d_B \times (M+N+C)}$ .

**Défis suivants** En observant les phrases « *white shirt with collar* » et « *shirt with white collar* », nous constatons que le marquage actuel attribuerait les mêmes étiquettes (« *white* : *B-color* » et « *collar* : *B-part* ») aux deux phrases, ce qui ne permet pas de distinguer leur sens (voir la figure 5.2). Cela soulève une question importante sur l'efficacité de la définition actuelle des étiquettes pour capturer les nuances sémantiques.

Une approche potentiellement meilleure pourrait consister à modifier les étiquettes elles-mêmes. Si nous décidons de modifier la méthode de marquage, il faudrait alors évaluer si une méthode de combinaison flexible des étiquettes, où les étiquettes sont générées dynamiquement en fonction du contexte (par

exemple, une étiquette « *color* » et une étiquette « *part* »), serait plus efficace qu’une méthode utilisant des étiquettes prédéfinies et statiques (par exemple, une étiquette « *collar'scolor* »). La dernière pourrait simplifier le processus en utilisant des modèles tels que le BiLSTM mentionné précédemment (voir la figure 5.2), qui prennent en compte le mot précédent et le mot suivant lors du marquage de chaque mot, offrant ainsi un contexte supplémentaire qui pourrait améliorer la précision des étiquettes. Cependant, cette méthode multiplie le nombre d’étiquettes statiques et risque de manquer de la flexibilité nécessaire pour couvrir toutes les variations sémantiques possibles.

En revanche, si nous choisissons de prédire séparément les étiquettes de domaine et d’intention pour ensuite les combiner dynamiquement, la méthode actuelle de marquage peut être conservée. Toutefois, il est alors nécessaire d’ajouter un mécanisme supplémentaire permettant de modéliser la dépendance entre les différents *slots* annotés. Cette étape pourrait consister en un module *DST* (Dialogue State Tracking) qui gère les meilleures annotations sémantiques parmi les  $N$  candidats et prédit l’état correct en se basant sur elles. Dans certains travaux, les meilleurs  $N$  candidats sont considérés comme les  $n$ -grammes d’annotations délexicalisées du même ordre que les étiquettes de *slots* (M. HENDERSON, THOMSON et YOUNG 2014a,b). Par exemple, la phrase « *white shirt with collar* » est annotée en «  $\langle color \rangle shirtwith \langle part \rangle$  » et la phrase « *shirt with white collar* » est annotée en «  $shirtwith \langle color \rangle \langle part \rangle$  ». Avec les différents ordres des tokens délexicalisés, les  $n$ -grammes d’annotations seront différents. Par exemple, le bi-gramme «  $\langle color \rangle shirt$  » de la première phrase se distingue de celui de la deuxième phrase «  $\langle color \rangle \langle part \rangle$  ». Cela aidera le module *DST* à comprendre et à distinguer ces structures sémantiques.

D’un autre côté, si nous choisissons de ne pas traiter le problème de l’étiquetage comme une classification séquentielle des tokens, l’utilisation d’un score de similarité pourrait réduire les complications et rendre le problème plus facile à résoudre en permettant de comparer directement les séquences textuelles avec les paires « *slot-valeur* », comme introduit précédemment (voir la section 5.2.2.3). Cette méthode demande un dictionnaire des paires attributs prédéfini, ce qui nécessite un effort supplémentaire du travail de terminologie. Les meilleurs  $N$  candidats pourraient également être entrées du module *DST*.

De plus, une phrase peut mentionner plusieurs attributs et même plusieurs

intentions. Il serait préférable de mettre en place une classification multi-étiquettes pour traiter plusieurs éléments sémantiques au sein d'une même phrase. S'il y a plusieurs attributs, nous devons encore prendre en compte la compréhension de la dépendance entre ces attributs. Mais s'il y a plusieurs intentions, la complexité des combinaisons augmente considérablement et il faut identifier les  $N$  meilleurs candidats comme entrée du module *DST*.

Enfin, nous devons peser les avantages d'un traitement direct de ces défis au sein du module *NLU*, par rapport à leur délégation au module de suivi de l'état du dialogue (*DST*), que ce soit en deuxième position dans la chaîne de traitement ou dans une approche de bout en bout couvrant les tâches du *NLU*. Un traitement direct dans le *NLU* pourrait être plus rapide à mettre en œuvre et plus facile à ajuster, allégeant ainsi le travail du *DST*, où même un modèle basé sur des règles pourrait suffire. En revanche, un module *DST* qui utilise les  $N$  meilleures hypothèses comme entrée nécessiterait un modèle neuronal de suivi de confiance (*Neural Belief Tracker*, *NBT* en anglais). La troisième solution consiste à utiliser un modèle de bout en bout qui traite conjointement les tâches de *NLU* et de *DST*. Ce modèle prend les énoncés de l'utilisateur et l'historique du dialogue comme entrée, et prédit directement l'état actuel de l'utilisateur. Cette approche pourrait offrir une meilleure performance à long terme en intégrant tous les aspects du dialogue dans un cadre unifié.

**Étiquetage souple** Pour gérer les phrases pouvant contenir plusieurs *slots* ou intentions, nous pouvons utiliser l'étiquetage souple (*soft labeling*). Cette approche assigne un score de confiance à chaque étiquette possible, permettant ainsi une classification multi-étiquette.

Concrètement, on applique la fonction Sigmoid à chaque *logit* pour obtenir des scores indépendants, que l'on interprète comme des probabilités de pertinence pour chaque classe. Un seuil peut ensuite être utilisé pour sélectionner les étiquettes les plus pertinentes. La fonction de perte binaire logarithmique permet d'entraîner le modèle en maximisant les scores des étiquettes correctes et en minimisant ceux des étiquettes non pertinentes.

Comparée à la classification mono-étiquette basée sur la Softmax, l'étiquetage souple offre une plus grande flexibilité et facilite la capture de nuances dans les phrases comportant plusieurs intentions ou attributs.

5.2.2.5 Module *NLU* par *LLM*

Avec l'avènement des modèles de langage de grande taille (*Large Language Models*, LLMs), tels que GPT et ses variantes, une nouvelle approche pour la détection d'intention a émergé. Contrairement aux modèles traditionnels nécessitant une étape explicite d'apprentissage de représentation suivie d'un classifieur, les LLMs peuvent résoudre directement les tâches de *NLU* grâce à leurs capacités avancées de compréhension et de génération de langage naturel.

Prenons la détection d'intention comme exemple, les LLMs peuvent être exploités de deux manières principales :

- **Adaptation en aval (ajustement fin spécifique à la tâche)** : Le LLM est ajusté sur un corpus annoté pour la détection d'intention. Cette méthode (*fine-tuning*) repose sur une adaptation explicite des poids du modèle via une fonction de perte comme l'entropie croisée. La probabilité d'intention est calculée par :

$$P_{int} = \text{Softmax}(W_{int} \cdot \text{LLM}(U))$$

où  $W_{int}$  est une couche linéaire entraînable, et  $\text{LLM}(U)$  représente les états cachés extraits de la séquence  $U$ . Cette approche nécessite un jeu de données d'entraînement volumineux et permet d'exploiter les représentations riches des LLMs tout en les spécialisant pour une tâche donnée.

- **Prompting et utilisation directe sans ajustement fin** : Cette méthode exploite la capacité pré-entraînée du LLM sans ajuster ses poids. L'entrée utilisateur  $U$  est concaténée avec un prompt pour former  $\text{prompt} + U$ , qui est traité directement par le LLM :

$$P_{int} = \text{Softmax}(W_{int} \cdot \text{LLM}(\text{prompt} + U))$$

Les performances du modèle dépendent fortement de la qualité du prompt, qui peut inclure des templates explicites ou des exemples supplémentaires (*few-shot learning*).

Les deux approches, l'ajustement fin (*fine-tuning*) et le *prompting*, présentent des avantages et des limitations distincts. L'ajustement fin repose sur l'utilisation d'un ensemble d'entraînement annoté, permettant de produire des modèles spécialisés qui offrent des performances robustes sur des intentions spécifiques. Cependant, cette méthode manque de flexibilité face à des changements de domaine ou

d'intention, car elle nécessite un nouvel entraînement à chaque modification de la tâche.

En revanche, le *prompting* ne requiert pas d'ajustement explicite des poids du modèle et peut être utilisé directement avec un ensemble de test pour évaluer l'efficacité des formulations proposées. Cette approche est particulièrement adaptée pour explorer de nouvelles intentions ou s'adapter rapidement à des domaines variés grâce à une conception adéquate des prompts. Toutefois, ses performances sont sensibles à la qualité des prompts et peuvent être contraintes par les connaissances contenues dans le modèle pré-entraîné, ce qui peut engendrer des difficultés dans certains contextes complexes ou spécifiques.

Dans notre système, nous classifions les intentions des utilisateurs en deux catégories principales : celles qui nécessitent une recherche dans une base de données de produits (*true*) et celles qui n'en nécessitent pas (*false*). Voici les correspondances pour chaque intention courante :

- **true** : inform, find\_similar
- **false** : ask\_attr, ask\_opinion, compare, sort, chitchat

Pour effectuer cette tâche, nous utilisons la méthode du *prompting* en *few-shot*. Un prompt contenant des instructions et des exemples est construit pour guider le modèle. Voici un exemple complet de prompt utilisé :

**RÉPONDEZ UNIQUEMENT PAR « true » OU « false ».**

**Exemples de cas *true* :**

- « Avez-vous des chemises bleues ? »
- « Je veux acheter des chaussures Nike. »
- « Montrez-moi des robes pour les vacances d'été. »
- « Je cherche des vestes d'hiver. »
- « Y a-t-il des pulls rouges ? »
- « Vendez-vous des jeans ? »
- « Avez-vous quelque chose de similaire à cette robe ? »

**Exemples de cas *false* :**

- « Bonjour. »
- « Combien coûte ce t-shirt ? »
- « Quelle est votre politique de retour ? »
- « Comment puis-je suivre ma commande ? »



- « Merci. »
- « Que pensez-vous de la marque Dior ? »
- « Entre celui-ci et celui-là, lequel est le meilleur ? »
- « Trier par prix. »
- « Puis-je modifier mon adresse ? »
- « Quand ma commande arrivera-t-elle ? »
- « Comment retourner cet article ? »

**Directives :**

1. Retournez **true** si l'utilisateur :
  - Pose des questions sur des vêtements spécifiques.
  - Cherche à acheter ou trouver des vêtements.
  - Demande à voir des produits vestimentaires.
2. Retournez **false** si l'utilisateur :
  - Envoie des salutations ou des remerciements.
  - Pose des questions sur la disponibilité ou les prix des produits.
  - Trie des prix parmi des produits récupérés.
  - Pose des questions sur les livraisons, retours, politiques ou avis sur la mode.
  - Discute de l'état d'une commande.
  - Fait de la conversation légère.

### 5.2.3 Module DST

Jusqu'à présent, nous avons une compréhension générale de la structure de base du module *NLU* et des modèles couramment utilisés. Nous pouvons résumer les sorties du *NLU* comme suit :

1. Résultat de classification mono-étiquette et de l'étiquetage. Par exemple, pour la phrase « *I want to find a white dress* », l'intention est « *inform* », le domaine est « *dress* », et les étiquettes des *slots* est [*O, O, O, O, O, < B - color >, O*]. Ce résultat peut être exprimé sous forme de cadres sémantiques comme « *inform(domain = dress, color = white)* » ou sous forme délexicalisée comme « *I want to find a < color > dress* », puis utiliser cette phrase délexicalisée comme features d'entrée pour le module *DST*. Le *DST* profitera de l'information sous-jacente de

l'ordre des phrases pour distinguer les sémantiques subtiles (par exemple, « *shirtwith < color > < part >* » et « *< color > shirtwith < part >* »).

2. Résultat de classification multi-étiquette, notamment des étiquettes avec leurs scores de confiance respectives. Parfois, un seuil ou un nombre  $N$  est défini pour obtenir les  $N$  meilleures options.

Dans cette section, nous décrivons le module *DST* et expliquons comment il traite ces types d'informations fournies par le *NLU* et actualise l'état du dialogue à chaque tour de conversation.

### 5.2.3.1 Hypothèse et score de confiance

Pour un module *DST*, les hypothèses sémantiques pourraient provenir des sorties du module *NLU* ainsi que du message brut de l'utilisateur.

D'abord, nous voulons examiner la différence entre le *NLU* et le *DST*. Les deux modules sont étroitement liés, et tous les deux effectuent le remplissage des *slots* pour le dialogue comme une de leurs tâches. Cependant, ils jouent en réalité des rôles différents. Comme introduit dans la dernière section, le module *NLU* fait la classification de l'intention et du domaine pour le message actuel, ainsi que l'étiquetage des *slots* par token du message. Par exemple, étant donné un message utilisateur « *Find me a dress of Dior.* », le module *NLU* convertira le message brut en « *inform(domain = dress; brand = Dior)* » avec les étiquettes « *[O, O, O, O, O, < B – brand >]* ». Cependant, au contraire que la classification et l'étiquetage séquentiel, le module *DST* fait plutôt le remplissage de *slots* : Il tente de trouver une valeur pour chaque *slot* à partir d'une liste prédéfinie en fonction de toute l'historique du dialogue. Par exemple, il existe une liste prédéfinie « *intent : \_; domain : \_; brand : \_; priceRange : \_; color : \_; ...* », où le tiret bas derrière le deux-points est un espace réservé indiquant que cet emplacement peut être rempli avec une valeur. À chaque tour, le module *DST* examinera toute l'historique du dialogue jusqu'au tour actuel et décidera du contenu qui peut être rempli dans un *slot* spécifique de la liste. Si le message utilisateur « *Find me a dress of Dior.* » est le seul message dans un dialogue, alors la liste peut être remplie comme suit : « *intent : inform; domain : dress; brand : Dior; priceRange : None; color : None; ...* ». Selon les sorties du *NLU* résumées ci-dessus, nous considérons trois configurations possibles pour le module *DST*.

Premièrement, et de la manière la plus simple, lorsque les sorties du *NLU* sont des classifications mono-étiquette, le *DST* traduit les intentions, domaines et attributs identifiés par le *NLU* en états compréhensibles pour le système et met à jour l'état actuel en remplissant la liste des états objectifs. Par exemple, si le dialogue continue et que l'utilisateur répond « *I prefer white* », le *DST* remplit la liste prédéfinie avec ces nouvelles informations en vérifiant l'état précédent du dialogue : « *intent : inform; domain : dress; brand : Dior; color : white; priceRange : None; ...* ».

Deuxièmement, le *DST* reçoit plusieurs hypothèses au tour  $t$ , chacune avec son propre score de confiance calculé par les fonctions telles que Sigmoides comme décrit précédemment. Dans ce cas, il faut gérer les états du dialogue accumulés. Le *DST* analyse l'historique de la conversation et des états précédents pour déterminer les hypothèses pertinentes et remplir la liste d'états objectifs. Dans ce cas, la méthode la plus simple pour le remplissage de *slots* est de distinguer les *slots* d'information et les *slots* de requête. Dans ce cas, la méthode la plus simple pour le remplissage de *slots* est de distinguer deux catégories : les *slots* d'information, qui correspondent aux valeurs fournies explicitement par l'utilisateur (par exemple *couleur = rouge*), et les *slots* de requête, qui correspondent aux informations demandées par l'utilisateur au système (par exemple *prix = ?*).

Pour un *slot*  $s$ , l'ensemble de ses valeurs détectées est donné par :

$$V_s^* = \{v \in V_s \mid \mathbb{P}(s, v \mid x) \geq 0.5\} \quad (5.15)$$

Ici,  $\mathbb{P}(s, v \mid x)$  est le score de confiance de la paire  $(s, v)$  étant donné l'entrée  $x$ . Pour les *slots* informatifs, on ne retient qu'un seul meilleur résultat (celui avec le score de confiance le plus élevé). Pour les *slots* de requête, tous les *slots* dans  $V_{\text{req}}^*$  sont considérés comme ayant été demandés. Comme les *slots* de requête servent à modéliser les requêtes ponctuelles des utilisateurs, ils ne nécessitent pas de suivi de croyance d'un tour à l'autre.

### 5.2.3.2 Modélisation conjointe

Pour réduire la complexité des systèmes modulaires, certaines architectures cherchent à fusionner le *NLU* et le *DST* au sein d'un modèle de bout en bout. Dans cette configuration, l'entrée du *DST* peut être le message brut de

l'utilisateur pour le tour actuel ainsi que l'historique complet de la conversation. Dans cette section, nous nous présentons ces architectures et discutons ses principales caractéristiques.

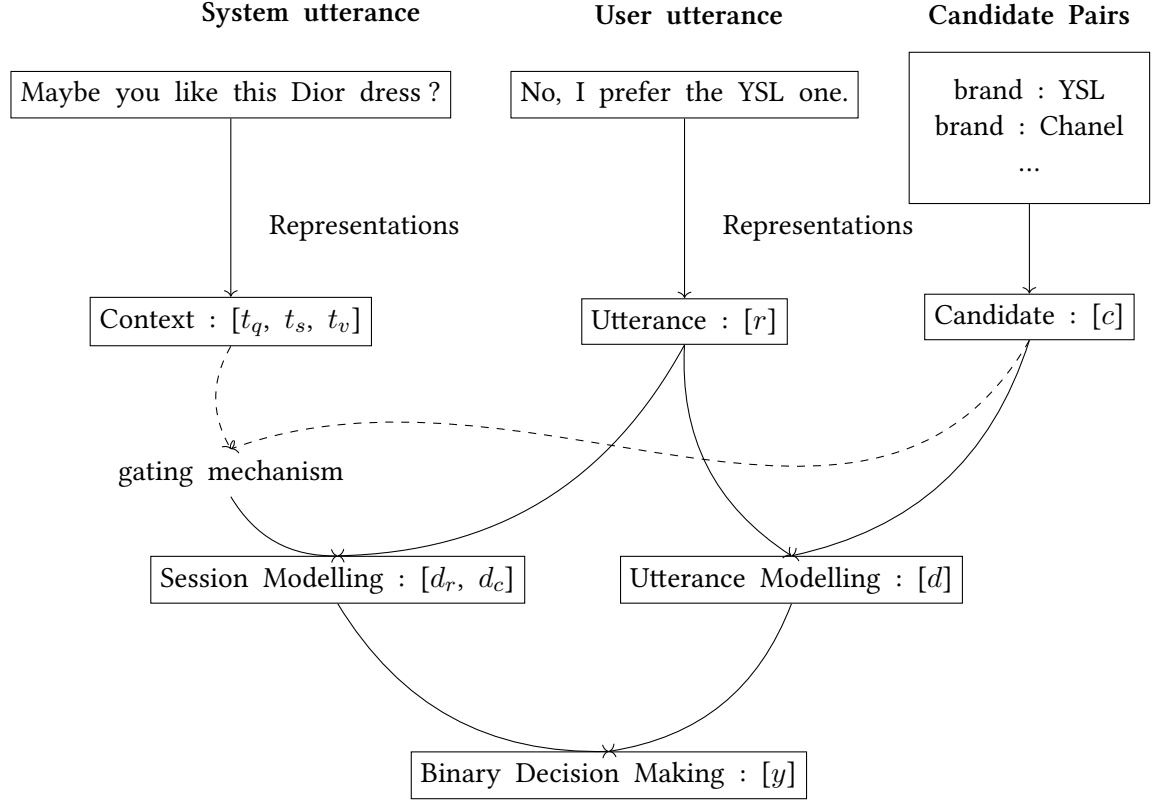


FIG. 5.3 : L'architecture du modèle de détection des *slots*. L'idée est de prédire si une paire *slot*-valeur candidate correspond au sens du dialogue en cours (décision binaire), en tenant compte à la fois de l'énoncé de l'utilisateur et des actes du système des tours précédents.

Bien que toutes les interventions précédentes du système et de l'utilisateur soient importantes, la plus pertinente est la dernière intervention du système (celle venue de la même session), dans laquelle le système de dialogue pourrait avoir effectué (parmi d'autres) l'un des deux actes suivants :

- Demande du système : Le système demande à l'utilisateur la valeur d'un *slot* spécifique  $T_q$ . Si l'énoncé du système est : « Quelle gamme de prix préférez-vous ? » et que l'utilisateur répond par n'importe laquelle, le modèle

doit déduire qu'il fait référence à la gamme de prix, et non à d'autres *slots* comme la zone ou le type de nourriture.

- Confirmation du système : Le système demande à l'utilisateur de confirmer si un couple *slot-valeur* spécifique  $(T_s, T_v)$  fait partie de leurs contraintes souhaitées. Par exemple, si l'utilisateur répond à « Que pensez-vous du design de Saint Laurent ? » par « Oui, je l'aime beaucoup », le modèle doit prendre en compte cette réponse afin de mettre correctement à jour l'état des croyances.

Ainsi, nous utilisons le dernier ensemble d'actes du système comme contexte, ainsi que l'énoncé de l'utilisateur, pour détecter une paire *slot-valeur* candidate dans l'état actuel du dialogue, comme illustré à la figure 5.3.

Soit  $t_q$  et  $(t_s, t_v)$  les représentations vectorielles de « requête », « slot » et « valeur » pour les actes du système (demande ou confirmation) comme décrits dessus (vecteurs nuls si aucun). Le modèle calcule les mesures de similarité suivantes entre ces actes du système, la paire candidate  $(c_s, c_v)$  et la représentation de l'énoncé  $r$  (voir les formules 5.4 et 5.5) :

$$\begin{aligned} d_r &= (c_s \cdot t_q)r, \\ d_c &= (c_s \cdot t_s)(c_v \cdot t_v)r \end{aligned} \tag{5.16}$$

Où  $\cdot$  représente le produit scalaire et les représentations vectorielles de  $t_q$ ,  $(t_s, t_v)$  et  $(c_s, c_v)$  partagent la même dimension, puisque le produit scalaire donne un scalaire comme résultat, les vecteurs  $d_r$ ,  $d_c$  et  $r$  sont également de même dimension. Ces « scores de similarité » calculés par le produit scalaire agissent comme des mécanismes de passage (*gating mechanism*) qui ne transmettent la représentation de l'énoncé que si le système a demandé spécifiquement le *slot* candidat ou la paire *slot-valeur* candidate actuelle. Ce type d'interaction est particulièrement utile pour l'acte de confirmation du système : si le système demande à l'utilisateur de confirmer, il est probable que l'utilisateur ne mentionne pas de valeurs de *slot*, mais réponde simplement par l'affirmative ou la négative. Cela signifie que le modèle doit considérer l'interaction entre l'énoncé de l'utilisateur, la paire *slot-valeur* candidate et la paire *slot-valeur* offerte par le système. Et seulement si ces deux dernières sont identiques, le modèle doit prendre en compte la polarité affirmative ou négative de l'énoncé de l'utilisateur

lors de la prise de décision binaire suivante. Enfin, ces deux représentations de résumé de la modélisation du contexte sont transmises au module de prise de décision, qui les combine avec la sortie du décodage sémantique pour prendre la décision finale. Nous appelons cette partie « modélisation de session » car l'énoncé du système et celui de l'utilisateur ensemble composent la session au temps  $t$ .

Pour la partie de la détection des *slots* du module *NLU*, nous l'appelons « modélisation de l'énoncé de l'utilisateur ». Dans la section précédente sur la *NLU*, nous avons présenté des méthodes pour identifier les *slots* dans un énoncé sans utiliser l'étiquetage séquentiel des *slots* (qui est essentiellement une classification séquentielle). Par exemple, on peut utiliser un dictionnaire externe de *slots* et de valeurs pour calculer un score de similarité avec l'énoncé, ou bien utiliser l'étiquetage souple pour calculer un score de confiance à chaque attribut (comme  $P(\text{color} = \text{white})$ ). Ici, comme nous voulons combiner les modules *NLU* et *DST* pour effectuer la détection des *slots*, nous pouvons intégrer ces deux approches.

Concrètement, soient les représentations dans l'espace vectoriel du nom de *slot* et de sa valeur d'une paire candidate, données par  $c_s$  et  $c_v$  (avec les vecteurs de mots des *slots*-valeurs multi-mots additionnés). Notre modèle apprend à faire correspondre cette paire vectorielle en un seul vecteur  $c$  de la même dimension que la représentation de l'énoncé  $r$ . Comme d'habitude, la fonction sigmoïde contraint les valeurs de  $c$  (les *logits*) dans l'intervalle 0-1. Ces deux représentations sont ensuite forcées d'interagir afin d'apprendre une métrique de similarité qui distingue les interactions des énoncés avec les paires *slot*-valeur qu'ils expriment ou n'expriment pas :

$$\begin{aligned} c &= \sigma(W_{sc}(c_s + c_v) + b_{sc}) \\ d &= r \odot c \end{aligned} \tag{5.17}$$

Où  $\odot$  représente la multiplication élément par élément (ou multiplication Hadamard) entre deux vecteurs ou matrices de même dimension. Bien que le produit scalaire que nous avons introduit avant et la multiplication élément par élément puissent tous les deux être considérés comme des moyens de mesurer la similarité entre deux vecteurs, ils présentent des différences subtiles. Le produit scalaire, qui peut sembler être la métrique de similarité la plus intuitive, réduit l'ensemble

riche de caractéristiques de  $d$  à un seul scalaire. En revanche, la multiplication élément par élément conserve les interactions de chaque dimension, offrant au réseau en aval la possibilité d'exploiter ces informations via ses couches non linéaires, et donc de modéliser des relations plus complexes entre  $r$  et  $c$ . Ce réseau (décision binaire dans la figure 5.3) utilise une couche cachée intermédiaire de taille 100 pour prendre la décision finale concernant la paire candidate actuelle.

### 5.2.3.3 Suivi d'états

Une fois que nous avons obtenu les informations ci-dessus (intentions, domaine, *slots*) et que nous disposons d'une base de données de produits, nous pouvons alors relier cette partie du travail à celle de la section 4.3.3 « génération de données ». Bien que nous ayons déjà introduit dans le chapitre précédent comment nous avons construit un simulateur de shopping et de fournisseur pour des robes, nous allons ici, du point de vue du système, élargir le domaine et explorer comment un agent conversationnel traite ces informations.

Le composant de suivi d'état maintient l'état du dialogue  $S_t$ , qui désigne la représentation de la session de dialogue jusqu'au temps  $t$ . Le suivi d'état fonctionne comme suit :

1. Met à jour l'état d'intention  $M_t.I$  en fonction de la fonction  $\text{IntentUpdate}(M_t, S_{t-1})$  qui met à jour l'intention de  $M_t$  selon les règles suivantes :
  - Si  $M_t.I \in I$  est une intention indépendante du contexte de session, alors  $M_t.I$  reste inchangé ;
  - Si  $M_t.I \in I$  est une intention dépendante du contexte de session, et  $S_{t-1}.I = \text{Inform}$ , alors  $M_t.I$  est défini comme *Inform* ;
  - Sinon,  $M_t.I$  est défini comme *Chit-chat*.

Ici, une intention dépendante du contexte de session est une intention dont la mise à jour repose sur l'historique du dialogue (par ex. l'intention du tour précédent).
2. Si une énonciation est détectée comme chit-chat, alors aucune mise à jour n'est effectuée,  $S_t = S_{t-1}$ .
3. Si la catégorie de produit de l'énonciation actuelle est identique à la catégorie de produit stockée dans  $S_{t-1}$ , alors  $S_t$  héritera de toutes les informations stockées dans  $S_{t-1}$ . Sinon, le contenu de  $S_t$  sera totalement

mis à jour en fonction de  $M_t$ , y compris le domaine ( $M_t.D$ ), l'intention ( $M_t.I$ ) et les *slots*-valeur ( $M_t.SV$ ).

Notez que, si le nombre d'énonciations chit-chat continues dépasse un nombre prédéfini, ou si l'intervalle de temps entre deux énonciations consécutives dépasse une durée prédéfinie à l'instant  $t$ , alors  $S_t$  sera effacé, comme un mécanisme d'oubli.

#### 5.2.3.4 Intégration des connaissances

Dans la section 4.2, nous avons catégorisé les connaissances en quatre types principaux : **catalogue**, qui représente des informations structurées sur les produits de mode ; **terminologie et taxonomie**, qui regroupent des connaissances structurées propres au domaine de la mode ; **persona**, qui inclut des caractéristiques et informations des utilisateurs, structurées ou non ; et enfin, les **connaissances contextuelles**, qui concernent des informations non structurées et de fond sur le domaine de la mode.

Les modèles traditionnels montrent une meilleure performance lorsqu'ils sont associés à des données structurées, car ces dernières nécessitent des traitements préalables rigoureux et une normalisation manuelle. En revanche, les grands modèles de langage (*LLM*) se distinguent par leur capacité à comprendre et à générer du contenu à partir de données non structurées. Ils peuvent également intégrer des informations structurées, offrant ainsi une flexibilité unique.

Dans la conception de notre module de suivi d'états (*DST*), nous avons donc adopté une approche hybride : les modèles traditionnels sont principalement utilisés pour exploiter les données structurées, tandis que les *LLMs* sont conçus pour traiter à la fois des données structurées et non structurées. Cette combinaison permet d'optimiser la gestion des connaissances, en capitalisant sur les forces spécifiques de chaque type de modèle et en minimisant les efforts requis pour le traitement préalable des données.

##### Modèles traditionnels avec catalogue :

Dans la section précédente, nous avons introduit la reconnaissance d'intention et le *slot filling* dans le cadre du NLU. En utilisant des modèles traditionnels, nous obtenons l'intention utilisateur  $M_t$ , qui inclut le domaine ( $M_t.D$ ), l'intention ( $M_t.I$ ) et les *slots*-valeur ( $M_t.SV$ ). Lorsque leur score de confiance dépasse un certain seuil, ces informations sont considérées comme valides et utilisées



pour interroger directement la base de données afin de trouver des produits correspondant aux critères ou des attributs spécifiques des produits.

Par exemple, supposons qu'un utilisateur exprime la requête suivante : « *Je cherche une robe rose pour une soirée.* » Le modèle traditionnel identifie :

- Domaine ( $M_t.D$ ) : *robe*.
- Intention ( $M_t.I$ ) : *Inform.*
- Slots-valeurs ( $M_t.SV$ ) : {"color" : "rose", "occasion" : "soirée"}.

Ces informations sont ensuite utilisées pour interroger le *catalogue* de produits et retourner une liste de robes roses adaptées pour une soirée.

L'avantage principal de cette approche est qu'elle repose exclusivement sur des règles explicites, ce qui minimise les besoins en calcul et en ressources de traitement. Cependant, cette méthode présente également certaines limites. Par exemple, le domaine identifié ( $M_t.D$ ) doit impérativement correspondre aux domaines prédéfinis dans le *catalogue*, ou au moins y être relié de manière explicite. De plus, les slots-valeur ( $M_t.SV$ ) identifiés doivent exactement correspondre à la terminologie utilisée dans le *catalogue*. Ainsi, si un utilisateur recherche une robe de couleur *pink* (en anglais), mais que certaines robes dans le *catalogue* sont étiquetées comme *rosé*, celles-ci risquent d'être exclues des résultats. Bien que *rosé* soit une nuance de *pink*, l'absence de correspondance explicite entre les termes entraîne du silence dans la recommandation des produits.

Pour remédier à ce problème, nous avons introduit un dictionnaire terminologique construit manuellement, qui recense les relations simples de correspondance et d'inclusion entre les termes. Ce dictionnaire permet de combler les écarts entre les mots utilisés par les utilisateurs et ceux présents dans le *catalogue*. Par exemple, pour les couleurs, le dictionnaire peut inclure une hiérarchie et des correspondances comme suit :

TAB. 5.3 : Exemple d'un dictionnaire terminologique pour les couleurs

Main Color	Variants and Synonyms
Red	{red, light red, dark red}
Light Red	{pink, rose, sakura, light pink, fluorescent pink, magenta}
Blue	{blue, sky blue, navy blue}
Green	{green, light green, dark green}
Yellow	{yellow, pastel yellow, gold}

**LLMs avec des connaissances non-structurées :**

Pour intégrer des connaissances non-structurées, comme les articles provenant de Wikipedia et Vogue, nous effectuons un pré-entraînement léger sur la version de base d'un *LLM*. Cette approche vise à adapter le modèle pré-entraîné sur des données générales à des connaissances spécifiques au domaine de la mode, tout en utilisant la méthode QLoRA pour réduire les exigences en matière de calcul.

L'objectif principal du pré-entraînement léger est d'affiner les représentations existantes du modèle en exploitant des données textuelles supplémentaires. Supposons que le modèle de base  $LLM_{base}$  ait été initialement pré-entraîné sur une vaste collection de données générales  $\mathcal{D}_{gen}$ , nous adaptons ses paramètres à un nouveau corpus  $\mathcal{D}_{fashion}$ , contenant des articles spécifiques au domaine.

Pour une séquence textuelle  $U = \{u_1, u_2, \dots, u_n\}$ , le modèle apprend à maximiser la probabilité conditionnelle suivante, décomposée auto-régressivement :

$$P(U) = \prod_{i=1}^n P(u_i \mid u_1, u_2, \dots, u_{i-1}; \theta)$$

Où  $\theta$  représente les paramètres du modèle. Pendant le pré-entraînement léger, seuls certains sous-ensembles des paramètres, tels que ceux ajustés par QLoRA, sont modifiés, tandis que le reste des paramètres du modèle de base reste figé.

Nous utilisons la fonction de perte standard entropie croisée, définie comme suit :

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^n \log P(u_i \mid u_1, u_2, \dots, u_{i-1}; \theta)$$

Bien que les catalogues structurés contiennent des informations précises sur certains produits, leur couverture reste souvent limitée. En revanche, presque chaque produit est accompagné d'un titre, qui constitue une source précieuse de connaissances non-structurées. Pour exploiter ces titres, nous avons recours à une méthode basée sur les embeddings (représentations vectorielles) afin de faciliter la recherche des produits les plus pertinents par rapport à une requête utilisateur.

Le processus consiste à convertir à la fois les titres des produits et les requêtes des utilisateurs en embeddings dans un espace de dimensionnalité réduite. Une fois ces embeddings générés, nous utilisons des techniques de recherche basée sur la similarité pour identifier les produits les plus proches de la requête dans

cet espace vectoriel. Pour obtenir des résultats plus précis, nous avons choisi d'affiner un modèle d'embedding existant, en l'adaptant aux spécificités de notre domaine. L'ajustement fin repose sur un ensemble de données annotées sous la forme de paires positives, où chaque paire associe une requête utilisateur à un produit pertinent (par exemple, une recherche de « Je veux trouver une robe rouge. » liée à un produit avec le titre « Robe élégante couleur bordeaux »).

Pour entraîner le modèle, nous utilisons la fonction de perte de classement à négatifs multiples (*Multiple Negatives Ranking Loss*), qui est particulièrement adaptée aux tâches de correspondance textuelle. Cette fonction maximise la similarité entre les embeddings des textes dans les paires positives tout en minimisant celle entre les textes non associés. Elle est définie comme suit :

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(q_i, p_i))}{\sum_{j=1}^N \exp(\text{sim}(q_i, p_j))},$$

Où  $q_i$  et  $p_i$  représentent respectivement les embeddings de la requête et du produit associés, et  $\text{sim}$  est une mesure de similarité (par exemple, produit scalaire ou cosinus).

### 5.3 Politique du dialogue : architecture d'auto-alignement

Dans le cadre de la recommandation de produits, la politique de l'agent peut être envisagée sous deux angles complémentaires : la **politique de recommandation** et la **politique de dialogue**.

Du point de vue de la **politique de recommandation**, un bon agent doit à la fois être capable d'identifier rapidement et avec précision les préférences explicites de l'utilisateur (exploitation), mais aussi d'explorer des pistes plus larges afin de proposer des produits susceptibles de susciter un intérêt inattendu (exploration), augmentant ainsi les opportunités de conversion. La gestion de ce compromis entre exploitation et exploration constitue donc un enjeu stratégique.

Du point de vue de la **politique de dialogue**, l'objectif est de guider le système dans le choix de comportements interactifs appropriés, tels que continuer à poser des questions de clarification, fournir directement une recommandation, ou encore engager une séquence de conversation informelle, afin à la fois de réduire progressivement l'espace des recommandations possibles et d'améliorer l'expérience utilisateur.

Dans les approches traditionnelles, l'apprentissage de ces politiques repose souvent sur des méthodes de renforcement ou d'apprentissage supervisé qui établissent une correspondance entre états de dialogue et actions, ou bien sur l'ajout d'un module de recommandation basé sur le filtrage collaboratif permettant de proposer à un utilisateur les produits préférés par des profils similaires.

Dans notre travail, en revanche, l'utilisation d'un grand modèle de langage comme modèle de base simplifie une grande partie de ces choix stratégiques. En particulier, de nombreux comportements relevant traditionnellement de la politique de dialogue peuvent être obtenus directement via un simple *prompting*, par exemple en fixant la personnalité de l'agent (voir la section 4.3.4.2). Quant à la politique de recommandation, au lieu d'ajouter un module séparé et d'exploiter des algorithmes classiques basés sur des données utilisateur-produit externes, nous proposons une approche novatrice reposant sur une architecture d'**auto alignement**. Celle-ci vise à aligner le modèle sur l'utilité utilisateur : d'une part en simulant les décisions de l'utilisateur (achat, ajout-au-panier, ou abandon), et d'autre part en interprétant ses intentions profondes pour transformer une requête utilisateur en une description de produit formulée dans un langage plus proche du discours commercial. Cette description est ensuite utilisée pour faire correspondre la requête à des produits réels, ce qui améliore la précision des recommandations et augmente le taux potentiel de conversion.

### 5.3.1 *Simulateur d'utilisateur*

Nous présentons le processus global de la construction du modèle du simulateur d'utilisateur que nous appelons « *Oracle* » dans la figure 5.4.

Le modèle Oracle est entraîné pour aligner les comportements utilisateurs à partir de notre jeu de données e-commerce à grande échelle (illustré dans la partie inférieure de la figure). L'objectif est de prédire les actions des utilisateurs (achat, ajout au panier, abandon) à partir des requêtes utilisateur (comprenant les requêtes en langage naturel et les clics historiques), du produit actuel, ainsi que de ses descriptions et avis, simulant ainsi le processus de prise de décision d'un utilisateur (partie supérieure de la figure).

Afin d'adapter la tâche au mode d'entraînement des grands modèles de langage et de les affiner pour en faire un simulateur de comportement utilisateur, nous reformulons la tâche de classification de la prédiction du comportement de l'uti-

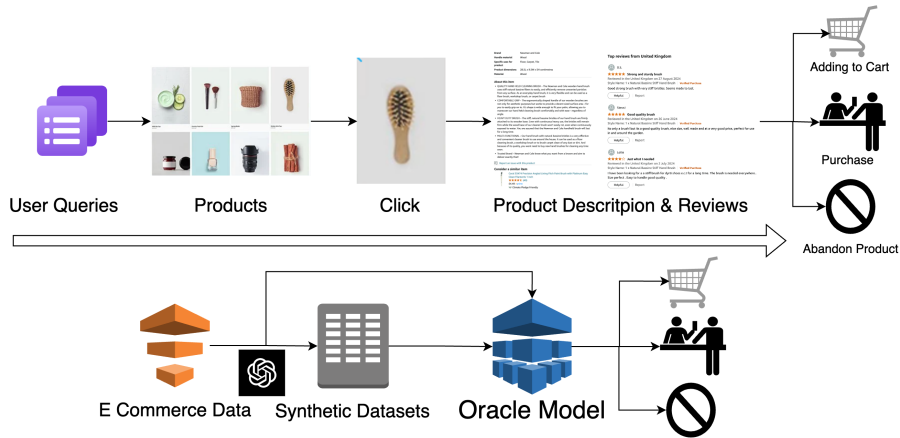


FIG. 5.4 : Comportement utilisateur (moitié supérieure) et logique d'entraînement et structure du modèle Oracle. Le modèle Oracle est basé sur Qwen2-72B et les données synthétiques sont générées par GPT-4o. Les choix d'action finaux de l'utilisateur sont : ajouter au panier, acheter ou abandonner le produit.

lisateur en une tâche de génération « text-to-text ». Les données d'entraînement sont intégrées dans un prompt. Par exemple, nous définissons le prompt de la manière suivante :

You are a customer and you are shopping online now. You have given a query: {QUERY} and you clicked {CLICK}, after the click, you have got product descriptions as: {PRODUCT DESCRIPTION}. You now take a decision, choose your answer among "buy", "add-to-cart" and "abandon".

Nous avons affiné et comparé plusieurs modèles de langage populaires, et nous présentons les résultats dans la section 6.4.

### 5.3.2 Générateur du contenu produit

Dans la section 5.2.3.4, nous avons décrit notre méthode d'intégration des connaissances non structurées dans le système. Celle-ci repose sur l'utilisation d'un modèle d'embedding pour calculer la similarité entre les contenus. Ce modèle d'embedding peut être soit un *LLM* purement textuel, soit un modèle multimodal. Une telle approche permet d'alléger la charge du grand modèle de base en déléguant l'action de « recherche » au module de gestion d'état, séparé des modules de compréhension du langage, de politique de dialogue et

de génération de langage.

Cependant, comme le modèle d'embedding et le grand modèle de base utilisé pour le dialogue ne sont pas identiques (le premier n'ayant pas de fortes capacités de compréhension et de génération), et étant donné le décalage naturel entre la formulation des requêtes des utilisateurs et la présentation des produits (titres, descriptions, voire images), il est nécessaire de combler ce fossé. Pour cela, nous introduisons, après le module *NLU* et avant la recherche par embedding, un module supplémentaire de *content generator*.

Ce module, qui partage le même *LLM* de base que le simulateur d'utilisateur (« Oracle »), a pour rôle de générer, à partir de la requête utilisateur, un contenu produit (titre ou description) cohérent avec l'intention exprimée. Ce contenu est ensuite évalué par le simulateur d'utilisateur pour décider d'un éventuel achat. On peut ainsi considérer ce module comme un « vendeur », formant avec le simulateur d'utilisateur (le « client ») une architecture d'auto-alignement (*self-alignment*).

L'intérêt de ce mécanisme est double : d'une part, les contenus générés s'approchent davantage du discours commercial et peuvent être considérés comme des produits « attendus » ; d'autre part, ils servent de pont entre la requête de l'utilisateur et la présentation effective du catalogue (titres et descriptions réels), faisant le calcul de la similarité d'embedding plus raisonnable et améliorant ainsi la qualité des recommandations.

Prenons un exemple du scénario suivant :

**requête de l'utilisateur** : “I am going to host a large dinner party, and as the hostess I really need a proper outfit.”

**historique de navigation** :

- elegant evening dress, little black dress, fit, classic chic
- pearl necklaces 24mm Akoya pearl, white, strong light
- Limited edition Chanel 25 handbag, evening handbag

**titre généré** : “Elegant Evening Gown with Subtle Pearl Embellishments”

**description générée** : “refined full-length evening gown ; designed for formal gatherings ; delicate pearl details ; flattering silhouette ; formal, event, banquet, luxury ; old-money style.”

Étant donné que nous utilisons un simulateur utilisateur « Oracle » pour modéliser les décisions des clients, nous obtenons pour chaque contenu généré une évaluation explicite indiquant s'il correspond — ou non — à l'utilité du client. Par conséquent, contrairement à l'approche traditionnelle d'entraînement des *LLMs*, qui repose principalement sur la maximisation de la vraisemblance, nous proposons un schéma d'optimisation fondé sur l'utilité. La section suivante présentera en détail cette approche d'optimisation guidée par l'utilité.

## 5.4 Optimisation Kahneman-Tversky (KTO)

Dans cette section, nous introduisons une approche novatrice pour aborder les défis liés à l'optimisation des modèles de langage en utilisant la Théorie des Perspectives de Kahneman et Tversky (ETHAYARAJH et al. 2024), un concept issu de la psychologie. La Théorie des Perspectives offre un cadre plus réaliste pour modéliser le comportement humain par rapport à l'hypothèse de l'agent rationnel, en intégrant des concepts tels que l'aversion à la perte et l'inversion des préférences (TVERSKY et KAHNEMAN 1992). Plutôt que d'optimiser simplement la log-vraisemblance, nous appliquons une optimisation fondée sur l'utilité simulé par notre modèle « Oracle ».

### 5.4.1 Fonction de valeur de base dans KTO

Dans le cadre du modèle *KTO*, nous cherchons à ajuster les paramètres du modèle  $\theta$  afin de maximiser une fonction d'utilité qui intègre des biais de type humain. Une fonction de valeur  $v(z; \lambda, \alpha, z_0)$  associe à un résultat  $z$ , relativement à un point de référence  $z_0$ , la formulation suivante :

$$v(z; \lambda, \alpha, z_0) = \begin{cases} (z - z_0)^\alpha & \text{si } z \geq z_0 \\ -\lambda (z_0 - z)^\alpha & \text{si } z < z_0 \end{cases} \quad (5.18)$$

Dans cette fonction de valeur,  $z$  représente le résultat effectif ou le gain/perte ressenti par un individu, tandis que  $z_0$  désigne l'attente initiale ou le point de référence à partir duquel l'individu évalue ses gains ou ses pertes. Si  $z > z_0$ , la situation est perçue comme un gain ; sinon, elle est perçue comme une perte. Ce point de référence est donc essentiel, puisqu'il détermine si l'individu considère une situation comme favorable ou défavorable.

Cette fonction modélise le processus psychologique de prise de décision en situation d'incertitude, en particulier lorsque les individus sont confrontés à des gains et à des pertes. Les deux paramètres  $\alpha$  et  $\lambda$  jouent un rôle clé dans la forme et le comportement de cette fonction de valeur :

- $\alpha$  : Ce paramètre contrôle la courbure de la fonction de valeur et est directement lié à l'aversion au risque. Une valeur faible de  $\alpha$  indique que l'individu est plus sensible aux petites variations de gains ou de pertes. À l'inverse, une valeur élevée de  $\alpha$  implique un comportement plus linéaire, traduisant une moindre sensibilité aux petites variations, ce qui peut être interprété comme une tolérance accrue au risque.
- $\lambda$  : Ce paramètre détermine la sensibilité aux pertes, ou l'aversion aux pertes. Une valeur de  $\lambda > 1$  signifie que les individus accordent plus de poids aux pertes qu'aux gains de même ampleur. En d'autres termes, les pertes sont perçues comme plus significatives que des gains équivalents, ce qui reflète l'idée selon laquelle les individus cherchent plus fortement à éviter les pertes qu'à obtenir les gains.

#### 5.4.2 Adaptation de KTO au commerce électronique

Dans le contexte du commerce électronique, l'ensemble des contenus existants est produit par le vendeur. Ainsi, toutes les stratégies de *fine-tuning* existantes, en l'absence de régularisation explicite, sont considérées comme étant de la perspective du vendeur. Par conséquent, la fonction de valeur pour le vendeur est définie comme  $v(\cdot) = \cdot$ , ce qui revient à ajuster directement les paramètres du modèle en fonction du contenu.

Du point de vue de l'acheteur, le  $z_0$  représente ses attentes initiales avant l'interaction avec un produit. Nous définissons  $z_0$  dans notre cadre selon deux aspects :

1. La note moyenne ( $R_{\text{avg}}$ ) : les évaluations numériques (étoiles) fournies par les acheteurs précédents, normalisées dans l'intervalle  $[0, 1]$  ;
2. Les caractéristiques clés mises en évidence dans les avis ( $X_{\text{key}}$ ). À partir des avis, nous demandons au LLM de générer les trois principales qualités et les trois principales préoccupations (soit 6 caractéristiques au total) exprimées en langage naturel comme descriptions du produit.



Ainsi, notre  $z_0$  peut être défini comme :

$$\max \left( R_{\text{avg}}, \frac{1}{6} \sum_{x_i \in X_{\text{key}}} \mathbb{E}[KL(\pi_{\theta'}(y|x) || \pi_{\theta}(y|x_i))] \right) \quad (5.19)$$

Où  $KL(\cdot || \cdot)$  désigne la divergence de Kullback–Leibler (KL). La fonction  $\pi_{\theta}(y|x)$  correspond à la distribution de probabilité conditionnelle des sorties possibles  $y$  étant donné une entrée  $x$ , paramétrée par  $\theta$ . Ici,  $y$  peut être interprété comme les descriptions de produit générées par le modèle. La fonction  $\pi_{\text{ref}}(y|x)$  se réfère à la distribution de sortie du modèle de référence, qui dans ce cas est le modèle pré-entraîné original. La divergence KL mesure l'écart entre notre modèle et le modèle de référence, reflétant dans quelle mesure les descriptions générées divergent des descriptions originales du produit.

Une divergence KL élevée indique un écart significatif par rapport aux descriptions originales. Si l'utilisateur choisit d'acheter le produit malgré une telle divergence, cela peut être interprété comme une appréciation du changement — suggérant que la nouvelle description correspond mieux à ses attentes. Dans ce cas, la description générée et le produit sont considérés comme « désirables ».

Inversement, si l'utilisateur n'achète pas le produit après un changement important de description, cela implique que la description nouvellement générée n'a pas satisfait ses attentes et est donc « indésirable ». Lorsque les utilisateurs réagissent positivement à une description très différente, cela peut révéler que la description originale était biaisée ou insuffisante. Dans de tels cas, le modèle est encouragé à générer des descriptions plus objectives et précises. À l'inverse, si les utilisateurs rejettent des contenus divergents, cela suggère que le contenu original était déjà adéquat, et que le modèle doit être contraint afin d'éviter des écarts excessifs.

Ainsi, dans le cadre de KTO, nous proposons de reformuler l'équation 5.18 de la manière suivante :

$$v(x, y) = \begin{cases} \lambda_D \sigma(\beta(r_{\theta}(x, y) - z_0)) & \text{si } y \sim y_{\text{desire}} \mid x \\ \lambda_U \sigma(\beta(z_0 - r_{\theta}(x, y))) & \text{si } y \sim y_{\text{undire}} \mid x \end{cases} \quad (5.20)$$

Où  $r_{\theta}(x, y) = \log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)}$ , qui représente la différence entre les deux modèles en termes de probabilité de générer une description donnée  $y$  pour un même produit, et sert à mesurer la divergence entre les descriptions générées et

originales.  $\lambda_D$  et  $\lambda_U$  sont des hyper-paramètres associés respectivement aux pertes « désirables » et « indésirables ».  $\beta$  est un nombre compris entre 0 et 1 utilisé pour la régularisation. La fonction sigmoïde  $\sigma(\cdot)$  garantit que la fonction de valeur reste bornée et se comporte de manière lisse.

Enfin, la fonction de valeur globale est donnée par :

$$v(x, y) = v_{\text{buyer}}(x, y) + \gamma r_{\theta}(x, y) \quad (5.21)$$

Où  $\gamma$  est un facteur de régularisation.

**Troisième partie**

**Expérimentation**



# Chapitre 6

---

## Expérimentations

Dans ce chapitre, nous présentons en détail les expérimentations ainsi que les résultats obtenus. Notre système de dialogue couvre plusieurs tâches principales : la capacité de conversation de base (y compris les échanges généraux et les discussions portant sur des connaissances du domaine de la mode), la reconnaissance de l'intention utilisateur (*NLU*, voir la section 5.2.2), la recherche de produits, et l'alignement avec le comportement utilisateur. Pour chacune de ces tâches, nous décrivons les jeux de données utilisés ainsi que les configurations de l'entraînement des modèles correspondants.

### 6.1 Dialogues

#### 6.1.1 Composition du jeu de données

Afin d'entraîner notre modèle à maîtriser les capacités conversationnelles de base ainsi qu'à acquérir des connaissances fondamentales dans le domaine de la mode, notre jeu de données est structuré en deux volets principaux : *connaissances* et *dialogues*.

TAB. 6.1 : Sources et caractéristiques des données de catalogue

Source	Taille	Format	Composants
Amazon-MMD	220k	json, csv	titre, description, image
FashionGen	2k	json	titre, attributs structurés
Reddress.com	3k	json	titre, description
Shein.com	13k	json, csv	titre, attributs structurés

Comme nous avons introduit dans la section 4.2, la partie *connaissances* comprend tout d'abord un catalogue de produits de mode à grande échelle que nous avons collecté et nettoyé (voir le tableau 6.1). Ensuite, nous avons construit

un répertoire de terminologie pour le domaine de mode général en regroupant des définitions issues de plusieurs sources, notamment des catalogues, Wikipédia et des glossaires spécialisés (voir le tableau 6.2). En plus des domaines larges de la mode tels que les vêtements, les chaussures et les accessoires, nous avons également collecté 75 marques renommées de bijoux et de montres, 868 termes relatifs aux montres et 1135 termes relatifs aux bijoux (voir la section 4.2.2).

TAB. 6.2 : Aperçu de la base de terminologie du domaine de la mode

Type	Quantité	Exemples
Synsets spécialisés dans la mode	716	shirt, trouser, tuxedo, loafer, stilletoes, sunglasses, handbag, earring
Attributs de mode	52	color, pattern, material, brand, style, band color, lens material, price, wash-care
Attributs visuels de mode	45	color, pattern, material, style, neck, sleeves, length, closure
Attributs meta-informatif de mode	7	brand, price, wash-care
Valeurs des attributs de mode, nombre moyen par attribut	4200	...

Nous avons également constitué un corpus de textes informatifs et de connaissances contextuelles, en collectant environ 18 000 articles de *wikipedia.org* et 8 000 articles de *vogue.com*. Nous avons construit un petit jeu de données de 7 000 paires "question-contexte" et de 3 000 paires "question-réponse à choix multiple" par gpt-4-o, ce qui nous a permis d'obtenir un ensemble des données multi-tâche pour pré-entraîner un *LLM* et pour entraîner un modèle d'embedding de connaissances comme mentionné dans la section 4.2.3.

Enfin, pour générer les dialogues à l'aide des automates, nous avons conçu 20 *persona utilisateur* comme décrit dans la section 4.3.3.2, ce qui nous a permis de générer un total de 12,800 dialogues automatisés (voir la section 4.3.3).

Par ailleurs, afin de simuler une diversité de profils et de comportements conversationnels plus réalistes dans les dialogues générés par les *LLMs* (voir la section 4.3.4), nous avons construit 100 « persona utilisateur », chacun décrit en langage naturel et intégrant des informations variées telles que des informations de base, des préférences individuelles ainsi que des traits de personnalité liés au comportement d'achat.

TAB. 6.3 : Statistiques des dialogues générés

	DressDial	LLMDial
Méthode	Automate	LLM
Sujet	Robe	Apparence générale
Taille	12 880	30 000
Nombre de produits	4816	430 600
Nombre de <i>slots</i>	17	52
Nombre des valeurs des <i>slots</i>	605	218 400
Nombre moyen de produits mentionnés par dialogue	14.1	19.17
Nombre de persona d'utilisateur	20	100
Nombre moyen de tour de paroles par dialogue	8.75	10
Nombre moyen de mots dans les énoncés des clients	10.26	14.63
Nombre moyen de mots dans les réponses des assistants	8.54	60.38
Taille de vocabulaire (fréquence $\geq 4$ )	1383	26 422

Pour augmenter encore la diversité des dialogues, nous avons testé trois types d'instructions données aux *LLMs* :

1. Interroger un seul *LLM*, lui fournir les informations sur 40 produits, et générer 1,000 dialogues ;
2. Interroger deux *LLMs* sans fournir d'information produit, afin de générer 1,000 dialogues ;
3. Interroger deux *LLMs* avec les informations sur 40 produits, afin de générer 28,000 dialogues.

### 6.1.2 Évaluation et analyse des dialogues générés

Les statistiques relatives à ces dialogues générés sont présentées dans le tableau 6.3. *DressDial* correspond à des dialogues générés par les automates dans le domaine des robes de la plateforme Shein, tandis que *LLMDial* désigne des dialogues produits par un *LLM* à partir d'un sous-ensemble de données de produits du corpus MMD.

#### 6.1.2.1 Résultat DressDial

Nous avons formé quatre annotateurs pour évaluer 100 dialogues générés par les automates. Chaque critère est noté sur une échelle de 0 (qualité minimale) à 5 (qualité maximale). Ces notes traduisent une appréciation subjective de la

qualité des dialogues produits (voir le tableau 6.4).

Parmi les critères, la « Fluidité du dialogue » et « Esthétique du vêtement » sont évaluées globalement à l'échelle de l'ensemble du dialogue. Les autres critères sont notés à chaque tour de dialogue. En ce qui concerne la pertinence du produit, à chaque tour, le simulateur de vendeur recommande au maximum cinq produits (sous forme d'images). L'annotateur est invité à examiner chaque produit et à lui attribuer une note selon les règles suivantes :

- Si le produit est manifestement hors sujet, incohérent ou incorrect, il reçoit une note de 0 ;
- Sinon, il reçoit une note de 1.

La note finale pour cette dimension est calculée en divisant le total des points obtenus par le nombre total d'images évaluées (moyenne pondérée). Il est important de noter que si un même produit est recommandé plusieurs fois au cours d'un dialogue, il est noté à chaque apparition, et chaque image (y compris les répétitions) est comptée dans le dénominateur lors du calcul de la moyenne. Autrement dit, le nombre total d'images inclut les doublons.

TAB. 6.4 : Évaluation subjective des dialogues générés par les automates

	Fluidité (langage)	Fluidité (dia- logue)	Pertinence (produit)	Politesse (ven- deur)	Rationalité (client)	Esthétique
Moyenne	4.65	4.09	3.93	4.81	4.33	3.67
Écart-type	0.48	0.69	0.89	0.53	0.93	1.06
Kappa	0.54	0.39	0.54	0.63	0.43	0.09

Les résultats de l'évaluation subjective des dialogues générés par les automates sont présentés dans le tableau 6.4. On observe que la fluidité du langage (4.65) et la politesse du vendeur (4.81) obtiennent des scores moyens particulièrement élevés, ce qui suggère que les dialogues sont jugés naturels et courtois. En revanche, la pertinence des produits (3.93) et l'esthétique (3.67) sont légèrement inférieures, indiquant que certains dialogues peinent à maintenir une adéquation optimale avec les caractéristiques des articles et à produire un style discursif jugé agréable.

Du point de vue de la fiabilité inter-annotateurs, les coefficients de Kappa varient sensiblement selon les critères. On constate une bonne cohérence pour la



politesse (0.63) et la fluidité du langage (0.54), alors que la rationalité du client (0.43) et surtout l'esthétique (0.09) présentent une subjectivité beaucoup plus forte. Ces résultats mettent en évidence que certains critères, comme l'esthétique, sont difficilement objectivables et dépendent fortement de l'interprétation individuelle des annotateurs.

#### 6.1.2.2 Résultat LLMDial

Nous avons également invité 6 annotateurs formés à évaluer les dialogues générés par les *LLMs*, chacun examinant 100 dialogues sélectionnés aléatoirement, selon les critères suivants :

1. **Exactitude de l'objectif** : si les réponses du client correspondent à l'objectif d'achat défini ;
2. **Exactitude de la réceptivité** : si le client se comporte conformément au niveau de réceptivité spécifié ;
3. **Exactitude de la patience** : si le client se comporte conformément au niveau de patience défini ;
4. **Fréquence d'informations observées** : si les informations explicites issues du *persona* apparaissent dans l'énoncé ;
5. **Fréquence de connaissances observées** : si les connaissances implicites mais cohérentes avec le *persona* apparaissent dans l'énoncé.

Nous rappelons que, un *persona* utilisateur désigne une représentation fictive mais réaliste d'un utilisateur type. Il regroupe un ensemble de caractéristiques définies — telles que les informations sociodémographiques (âge, genre, localisation), les préférences d'achat, le style de conversation ou encore certains traits de personnalité. Ces *personas* peuvent être exprimés de manière concise (par quelques mots-clés comme patiente, réceptif, intéressé par la mode durable) ou sous la forme d'une phrase descriptive plus riche (« une jeune femme de 25 ans, passionnée par la mode locale et attentive au rapport qualité-prix »). Comme décrit dans la section 4.3.4, chez nous un *persona client* est défini comme un ensemble de caractéristiques comprenant :

1. **Informations de base** : âge, genre, zone de résidence, statut et centres d'intérêt ;
2. **Objectif d'achat** : type d'article recherché, taille, préférences de détails ;
3. **Style de conversation** : patience, réceptivité, type de demandes.

À partir de cette définition, nous distinguons deux types de contenus apparaissant dans les dialogues :

- **Informations observées** : il s’agit des cas où le client reprend explicitement dans la conversation un élément défini dans son *persona*. Par exemple, si le *persona* précise « You have a strong interest in exploring local places », et que le client dit « I’m interested in exploring local places », nous considérons que cette information a été observée.
- **Connaissances observées** : il s’agit des informations qui ne figurent pas directement dans le *persona*, mais qui peuvent être raisonnablement déduites à partir de celui-ci. Par exemple, si le *persona* indique un revenu de 0–30k dollars par an, et que le client formule la demande « Could you please sort these options by price from low to high », nous considérons que le simulateur exprime une connaissance dérivée de son profil.

Cette distinction fonde les deux derniers critères retenus pour l’évaluation par les annotateurs.

Chaque annotateur est invité à examiner si chaque critère est satisfait pour chaque énoncé, et à annoter 1 si c’est le cas, 0 sinon (voir l’annexe A.2). Nous faisons le rapport des résultats dans les paragraphes suivantes.

**Effet des personas utilisateur** Lorsque certains contenu du persona utilisateur sont modifiées (ajoutées ou supprimées) dans l’instruction du *LLM*, nous constatons que le dialogue généré peut changer de manière significative, par exemple en omettant des informations pertinentes. Les figures 6.1 et 6.2 illustrent bien les performances en termes de persistance des persona utilisateur. Le simulateur de client basique atteint déjà un taux très élevé (supérieur à 0,9) d’exactitude concernant l’objectif d’achat et la réceptivité. Par ailleurs, les études de variation apportent plusieurs enseignements concernant la conception des *user persona* :

1. Pour maintenir la cohérence de l’objectif d’achat du simulateur, il est important d’intégrer les personas *intérêt* et *revenu*. En revanche, inclure les personas *patience*, *taille*, *besoin* et *détail* peut nuire à cette cohérence.
2. Pour préserver la réceptivité du simulateur, celui-ci devrait être initialisé avec les personas *taille*, *besoin*, *intérêt*, *revenu* et *détail*, sans inclure le persona *patience*.
3. Pour maintenir la patience du simulateur, il est préférable d’ajouter le

persona *besoin* et de retirer le persona *intérêt*, afin de limiter les intentions secondaires.

4. Afin d'observer davantage d'informations de base et d'extrapolations de connaissances, il vaut mieux retirer le persona *patience*.
5. Pour favoriser l'extrapolation de connaissances à partir des personas, le simulateur utilisateur doit inclure un historique de consommation (*historique de consommation*).

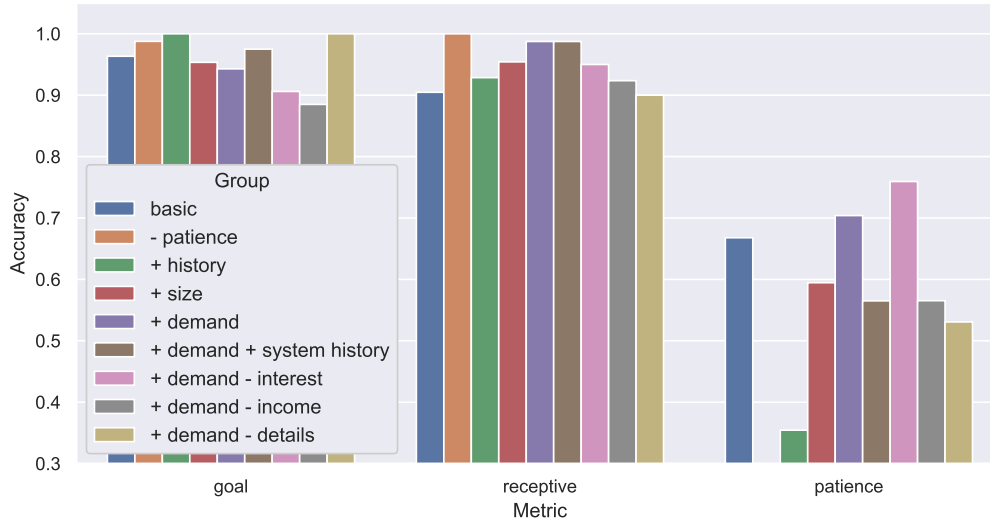


FIG. 6.1 : Cohérence entre l'objectif d'achat (*goal*) et les traits de personnalité du client (*receptive*, *patience*) avec ceux définis dans le *user persona*. Nous comparons la version complète (*basic*) aux expériences de variation.

**Effet de la politique de l'assistant de vente** Nous analysons également l'influence des politiques adoptées par les assistants de vente. Nous avons conçu quatre types de politiques qui guident les réponses de l'assistant face aux clients, afin de simuler différents styles d'interaction que les utilisateurs peuvent rencontrer (voir la section 4.3.4). En général, les assistants sont définis par 4 types :

- **Type 1.** Occupé : réponses courtes, recherche rapide sans bavarder.
- **Type 2.** Chaleureux : amical, pose des questions pour mieux comprendre le client.
- **Type 3.** Charmeur : bavard, recommande même au-delà des besoins exacts.

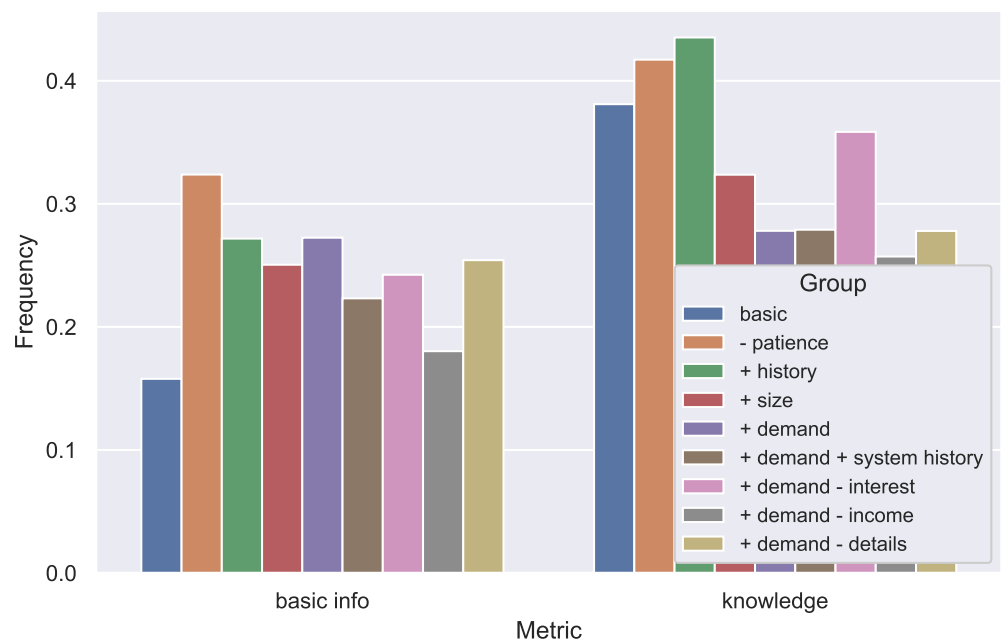


FIG. 6.2 : Fréquence des informations observées (*basic info*) et de l’extrapolation de connaissances (*knowledge*). Nous comparons la version complète (*basic*) et les expériences de variation.

– **Type 4.** Perdu : nul, poli mais inefficace, ne sait pas toujours quoi répondre.

Pour les clients avec différents niveaux de réceptivité, nous évaluons leur fidélité aux personas définis. Comme on peut le voir dans le tableau 6.5, l’exactitude est globalement élevée, sauf pour le Type 4 d’assistant. Cela indique que lorsque l’assistant est « stupide », le client a du mal à maintenir le niveau de réceptivité prévu, ce qui n’est pas rassurant et nous permet de confirmer que notre évaluation tient correctement compte des personas définies.

TAB. 6.5 : Exactitude de la réceptivité selon les différents types d’assistants de vente

	Type 1	Type 2	Type 3	Type 4
Réceptivité élevée	0.9333	1.0000	0.9815	0.7891
Réceptivité faible	0.9610	0.9505	0.9917	0.9375

De manière similaire, nous évaluons l’exactitude concernant la patience. Le

tableau 6.6 montre que les clients avec une patience élevée respectent généralement bien ce paramètre. Cependant, les autres groupes ont également tendance à se comporter de manière patiente. Une explication possible est que ChatGPT est conçu pour répondre avec patience, ce qui peut influencer le comportement du client simulé.

TAB. 6.6 : Exactitude de la patience selon les différents types d’assistants de vente

	Type 1	Type 2	Type 3	Type 4
Patience élevée	0.8823	0.8579	0.9563	0.8750
Patience moyenne	0.3633	0.6111	0.5951	0.7583
Faible patience	0.3681	0.5881	0.6238	0.6174

**Effet de l’historique de consommation** La figure 6.3 montre la comparaison des effets liés à l’ajout d’un historique de consommation du côté du client et du côté de l’assistant de vente. Les résultats suggèrent que l’ajout de l’historique du côté du client améliore l’exactitude de l’objectif d’achat, la fréquence d’apparition des informations de base, ainsi que l’extrapolation de connaissances. Par ailleurs, le fait de révéler l’historique de consommation à l’assistant de vente améliore la cohérence des personas en matière de réceptivité et de patience.

Dans l’ensemble, les évaluations humaines confirment que les dialogues générés — tant par la méthode par automates (DressDial) que par les *LLMs* (LLMDial) — produisent des échanges globalement cohérents et plausibles dans le cadre de l’achat de mode. Les annotateurs notent en particulier une bonne fluidité linguistique et une politesse des vendeurs, tandis que la pertinence produit et l’esthétique se révèlent plus variables. Pour la seconde méthode (LLMDial), l’évaluation de conformité aux personas montre que, dans la majorité des cas, les comportements des deux agents simulés reflètent effectivement leurs personas assignés. Ces résultats suggèrent que les modèles tiennent compte des profils simulés et se comportent, de façon générale, conformément aux attentes définies par nos scénarios. La section suivante détaille précisément la manière dont ces dialogues générés ont été exploités pour le pré-entraînement et l’affinage du modèle de dialogue (voir la section 6.1.3).

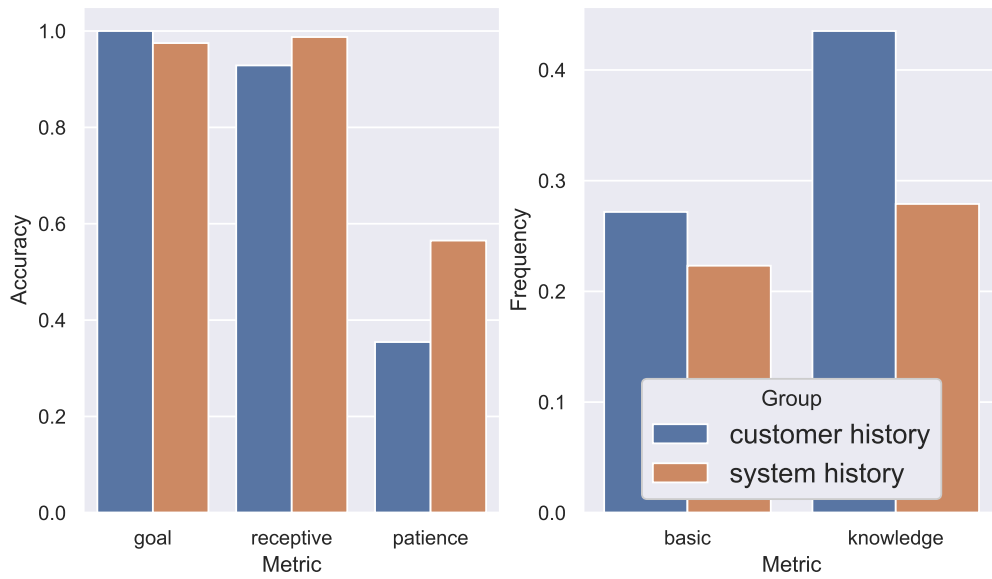


FIG. 6.3 : Effets de l'intégration de l'historique de consommation dans le dialogue. À gauche, la cohérence entre l'objectif d'achat (*goal*) et les traits de personnalité du client (*receptive*, *patience*) avec ceux définis dans le *user persona* ; à droite, la fréquence d'observation des informations de base (*basic*) et des connaissances extrapolées (*knowledge*). **Historique client** signifie que seul le client connaît son historique de consommation, tandis que **historique système** signifie que l'assistant de vente en est également informé.

### 6.1.3 Pré-entraînement léger et affinage du modèle

Comme nous avons introduit dans la section 5.1, nous distinguons deux étapes d'adaptation : un pré-entraînement léger (*lightweight pretraining*) sur des textes spécialisés afin d'ancrer le modèle dans le domaine de la mode, puis un affinage supervisé (*finetuning*) sur des dialogues générés pour l'adapter au format conversationnel et au respect des personas.

Nous avons d'abord testé le modèle Mistral-7B en dialogue avec des prompts spécifiques. Nous avons constaté qu'il possédait déjà certaines connaissances de base, mais avait tendance à se comporter comme un conseiller en style plutôt qu'un véritable assistant d'achat.

Afin d'intégrer davantage de connaissances spécifiques au domaine, nous avons effectué un pré-entraînement léger en utilisant un ensemble de don-

nées composé de 26 000 articles (dont 18 000 issus de Wikipedia et 8 000 de Vogue), en nous appuyant sur le framework Firefly (<https://github.com/yangjianxin1/Firefly>).

Nous avons utilisé la méthode de pré-entraînement QLoRA avec une taille de batch minimale, ce qui permet l'entraînement sur un seul GPU.

Les modèles testés incluent :

- Bloom 3B
- Qwen 7B
- Mistral 7B
- Qwen2 7B

Les améliorations observées restent limitées, bien que certaines connaissances génériques du domaine aient été apprises. Des cas d'utilisation détaillés sont présentés dans l'annexe A.3.

Ensuite, nous avons effectué un fine-tuning en une seule époque sur ces modèles avec les dialogues générés dont nous venons de décrire la création et l'évaluation (12k dialogues par automate et 28k dialogues simulés par GPT), sans information produit explicite.

L'entraînement a également été réalisé avec l'outil Firefly, en utilisant la méthode QLoRA et une taille de batch minimale, afin de permettre l'entraînement sur une seule carte GPU.

Nous avons observé des améliorations notables, en particulier que le modèle devient capable de « vendre » des produits. Des cas d'utilisation détaillés sont également présentés dans l'annexe A.3.

## 6.2 Module *NLU*

### 6.2.0.1 Composition du jeu de donnée

Le jeu de données NLU a été généré à l'aide de GPT-4o et contient un total de 1 110 exemples. Chaque entrée est constituée d'une requête utilisateur et d'une annotation sémantique sous forme d'intentions et d'entités (voir la section 5.2.2.1).

Un exemple typique d'annotation est le suivant :

```
{'uttr': "I'm looking for some stylish scarves like those from Hermes.  
Are they available in silk? Sort them by design."}
```

```

'nlu': [
  {'Domain': 'scarf', 'Intent': 'find_similar',
   'Slot': [{'name': 'brand', 'value': 'Hermes'}]},
  {'Domain': 'scarf', 'Intent': 'ask_attr',
   'Slot': [{'name': 'material', 'value': 'silk'}]},
  {'Domain': 'scarf', 'Intent': 'sort',
   'Slot': [{'name': 'design'}]}
]

```

Les annotations varient en fonction de la complexité de la requête utilisateur. Nous avons analysé la distribution du nombre d'intentions par exemple :

Nb d'intentions	Nb d'exemples
1	732
2	131
3	231
4	15
5	1

TAB. 6.7 : Distribution du nombre d'intentions

Intention	Nb d'exemples
inform	111
ask_attr	285
ask_opinion	263
find_similar	27
compare	26
sort	18
chitchat	2

TAB. 6.8 : Répartition des intentions parmi les 732 exemples qui n'ont qu'une seule intention.

Un sous-ensemble des données a été utilisé pour entraîner un classifieur binaire déterminant si un tour de dialogue nécessitera une recherche de produit. L'objectif est de filtrer les demandes ne relevant pas directement de la recherche d'articles vestimentaires. Nous avons pour cela considéré que seules les intentions *inform* et *find\_similar* doivent déclencher une recherche de produits. :

### 6.2.1 Affinage des modèles

Dans cette partie, nous présentons les résultats pour la classification multi-classe et binaire. Nous utilisons les 732 exemples annotés avec une seule instance *NLU* (c'est-à-dire un seul *domaine* et une seule *intention*) comme ensemble d'entraînement, et les 378 exemples restants, contenant plusieurs annotations *NLU*, comme ensemble de test. Pour ces exemples de test, une prédiction est



considérée comme correcte si au moins une des annotations *NLU* est correctement identifiée.

Le prompt utilisé pour la classification binaire est défini comme suit (voir l'annexe A.4 pour les instructions complètes) :

You are an agent that determines if user input requires querying a fashion product database (clothing, accessories, shoes, etc). Your task is to return a single boolean response: true if the input needs a product search among database, else false.

RESPOND WITH ONLY True OR False.

Les modèles pour la classification binaire testés incluent :

- Bi-LSTM
- Qwen2-7B
- LLaMA3-8B

Les résultats dans le tableau 6.9 montrent que le zéro-shot reste difficile pour tous les modèles, ce qui confirme la spécificité de la tâche. Un fine-tuning limité à quatre époques améliore néanmoins clairement les performances, en particulier pour les LLMs qui s'adaptent mieux que le Bi-LSTM. Le choix de n'entraîner que quatre époques s'explique par la taille limitée du jeu d'entraînement (732 exemples annotés) et par la volonté de réduire les risques de surapprentissage. Les scores finaux restant modérés, davantage de données annotées ou des techniques d'augmentation permettraient toutefois à un entraînement plus long de conduire à des gains supplémentaires.

TAB. 6.9 : F1 scores des modèles pour la classification d'intention dans *NLU* en zéro-shot et après 4 époques d'entraînement

	Classification binaire		Classification multi-classes	
	zéro-shot	4 epochs	zéro-shot	4 epochs
Bi-LSTM	0.474 (1 epoch)	0.562	0.385 (1 epoch)	0.491
Qwen2-7B	0.526	0.713	0.480	0.615
LLaMA3-8B	0.538	0.686	0.494	0.581

## 6.3 Module *DST* et Récupération de produits

Afin d'entraîner un modèle d'embedding optimisé pour une application de génération augmentée par récupération (RAG) dans le domaine du commerce en ligne, nous avons construit un jeu de données adapté et fait l'ajustement fin sur un modèle d'embedding. Compte tenu de la spécificité de notre scénario de dialogue, nous considérons deux configurations possibles pour la tâche de « recherche de produit à partir d'une requête ». La première consiste à utiliser directement l'énoncé de l'utilisateur comme requête pour retrouver les produits pertinents (voir la section 6.3.1). La seconde consiste à utiliser les informations du produit (titre, description ou image) comme requête (voir la section 6.3.2).

### 6.3.1 *Récupération entre énoncé et produit*

#### 6.3.1.1 Composition du jeu de données

Nous avons utilisé le jeu de données MMD (A. SAHA, KHAPRA et SANKARANARAYANAN 2017), un ensemble de conversations entre acheteurs et vendeurs, et avons extrait plus de 16 000 paires positives de requêtes et de produits recommandés. Plus précisément, nous avons appliqué les critères suivants pour la construction du jeu de données :

- La requête (*query*) est définie comme la première demande exprimée par l'acheteur dans chaque conversation. Cette phrase représente l'intention initiale de l'utilisateur et constitue un signal clé pour la récupération d'information.
- La réponse positive (*positive*) correspond au premier produit recommandé par le vendeur en réponse à cette requête.

En appliquant cette méthodologie, nous avons généré un ensemble de 16 000 paires (*query*, *positive product*). Ce jeu de données est conçu pour entraîner un modèle de recherche d'informations capable d'associer efficacement une requête utilisateur à un produit pertinent dans un environnement de commerce en ligne. Grâce à cette structuration, nous cherchons à améliorer la qualité des recommandations et à optimiser la pertinence des résultats lors de la recherche de produits.

### 6.3.1.2 Comparaison des modèles d'embedding de base

Avant d'entamer l'ajustement fin du modèle, il est essentiel d'évaluer les performances des modèles pré-entraînés existants afin d'établir une référence. Cette évaluation préliminaire nous permet de comprendre l'efficacité initiale du modèle et d'identifier les domaines nécessitant des améliorations.

Pour cette étude, nous avons utilisé les modèles `all-MiniLM-L6-v2` et `BAAI/bge-base-en-v1.5` comme modèles de base. Le dernier, doté de 109 millions de paramètres et d'une dimension cachée de 768, atteint un score de 63,55 sur le classement MTEB<sup>1</sup> (`muennighoff2022mteb`; ENEVOLDSEN et al. 2025), ce qui en fait l'un des modèles ouverts les plus performants pour sa taille.

L'évaluation a été réalisée à l'aide de « `InformationRetrievalEvaluator` », qui implémente des métriques d'évaluation de classement telles que « Mean Reciprocal Rank » (MRR), « Recall@k », « Mean Average Precision » (MAP) et « Normalized Discounted Cumulative Gain » (NDCG). Parmi ces métriques, le NDCG est considérée comme la métrique la plus pertinente.

Les résultats obtenus avec le modèle pré-entraîné sont les suivants :

Dimension	<code>all-MiniLM-L6-v2</code>	<code>BAAI/bge-base-en-v1.5</code>
768	–	0.0652
512	–	0.0591
256	0.0688	0.0552
128	0.0563	0.0397
64	0.0393	0.0293

TAB. 6.10 : Comparaison des performances NDCG@10 des modèles de base sans entraînement préalable (zero-shot). Le modèle `all-MiniLM` ne prend pas en charge des dimensions supérieures à 384 ; néanmoins, à 256 dimensions et au-dessous, il offre de meilleures performances que `bge-base`.

Ces résultats serviront de base de comparaison pour évaluer les améliorations apportées par l'affinage du modèle. Comme attendu, les scores rapportés dans le tableau demeurent assez faibles. Cela s'explique principalement par deux éléments : d'une part, les modèles considérés sont exploités en configuration *zero-*

<sup>1</sup>Le classement MTEB (Massive Text Embedding Benchmark) est un benchmark de référence qui évalue la qualité des représentations textuelles produites par les modèles de langage sur un large éventail de tâches (recherche d'information, classification, clustering, similarité sémantique, etc.)

*shot*, sans affinage préalable sur des données du domaine de la mode ; d'autre part, l'indicateur choisi, le NDCG@10, est particulièrement exigeant puisqu'il évalue la qualité du classement uniquement sur les dix premiers résultats parmi un ensemble de plus de 16 000 produits. Dans ce contexte, il est néanmoins intéressant de constater deux points saillants : (i) la dimension des embeddings a un effet notable sur les performances, et (ii) pour les différentes tailles testées, *all-MiniLM* obtient systématiquement de meilleurs résultats que *bge-base*.

### 6.3.1.3 Affinage des modèles d'embedding

Nous constatons que les modèles d'embeddings non affinés obtiennent un score NDCG très faible sur les 16 000 paires « requête - produit positif ». Une première hypothèse est que la métrique NDCG<sub>10</sub> est trop exigeante ; une autre est qu'un entraînement sur plusieurs epochs supplémentaires pourrait améliorer les résultats. Par ailleurs, il existe un écart linguistique important entre les requêtes formulées par les utilisateurs lors du dialogue et les intitulés des produits, ce qui rend la recherche de produits pertinents par simple similarité d'embeddings particulièrement difficile. Dans cette section, nous remplaçons la métrique de NDCG@10 par NDCG@100, et nous affinons les modèles pendant plusieurs epochs.

L'ajustement fin du modèle d'embedding a été réalisé en utilisant la technique d'apprentissage de représentation Matryoshka (KUSUPATI et al. 2022). Cette approche permet de créer des embeddings pouvant être tronqués à diverses dimensions sans perte significative de performance. En concentrant les informations essentielles dans les premières dimensions de l'embedding, le modèle gagne en efficacité tout en conservant une précision élevée dans les tâches en aval telles que la récupération d'informations, la classification et le regroupement.

Les résultats du modèle *all-minilm* sont présentés dans le tableau 6.11. L'entraînement a été effectué sur des GPU grand public (16 Go), tels que le NVIDIA A10G ou le RTX 4080, et peut être adapté à des GPU plus puissants si nécessaire. Les résultats obtenus après l'ajustement fin montrent une amélioration notable des performances en utilisant seulement 16 000 échantillons pour l'entraînement. De plus, l'utilisation de la technique MRL a permis de réduire la taille des embeddings de 6 fois tout en conservant plus de 99 % des performances, ce qui est particulièrement bénéfique pour les applications où les ressources de stockage

Dimension	Base NDCG@10	4 epochs NDCG@10	4 epochs NDCG@100	10 epochs NDCG@100	30 epochs NDCG@100	↑ (%)
256	0.0688	0.0818	0.1382	0.1514	0.1746	26.33
128	0.0563	0.0756	0.1279	0.1426	0.1729	35.18
64	0.0393	0.0597	0.1067	0.1331	0.1599	49.86

TAB. 6.11 : Comparaison des performances du modèle **all-minilm** pendant plusieurs epochs avec la métrique NDCG. Nous constatons une grande amélioration en comparant les résultats de NDCG@100 après l'affinage de 30 epochs et de 4 epochs.

et de calcul sont limitées.

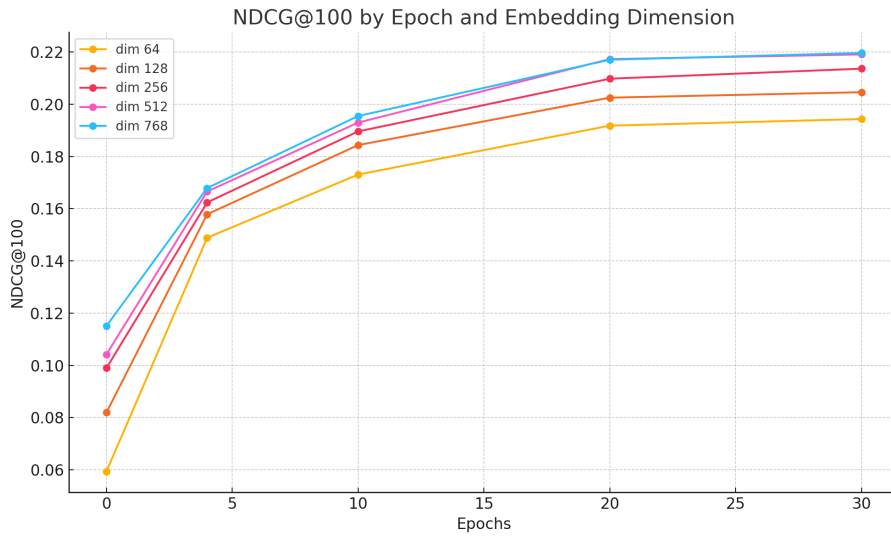
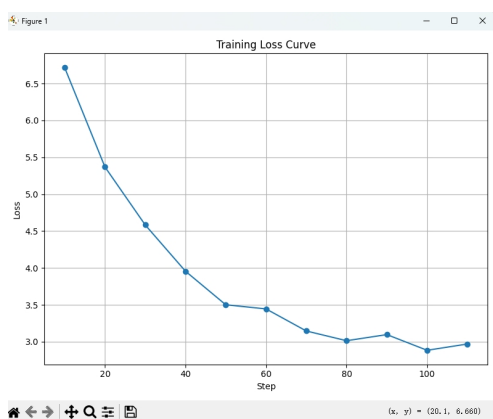
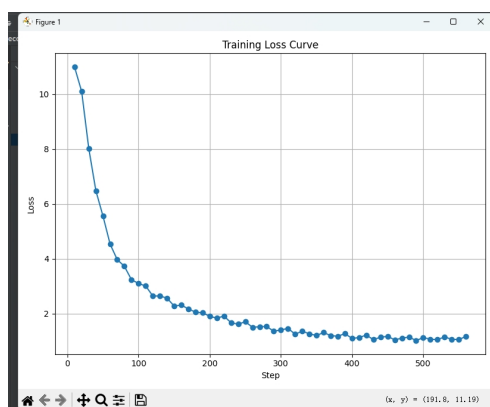


FIG. 6.4 : Performance du modèle **bge-base** sur NDCG@100 selon différentes époques d'entraînement et dimensions d'embedding.

Nous consolidons ces résultats en reprenant le modèle **bge-base**, qui nous permet de faire des expériences avec un plus grand nombre de dimensions. Les figures 6.4 et 6.5, démontrent l'efficacité du fine-tuning des modèles d'embedding pour des applications RAG dans le scénario de la recherche du produit relevant. Nous observons une amélioration rapide des performances de récupération au cours des 20 premières epochs, suivie d'une stabilisation progressive entre la 25<sup>e</sup> et la 30<sup>e</sup> epoch, indiquant que le modèle a déjà atteint des performances quasi optimales.



(a) Courbe de loss et NDCG@10 pour bge-base



(b) Courbe de loss et NDCG@100 pour bge-base

FIG. 6.5 : Courbes de loss et de performance (NDCG) pour le modèle **bge-base**, illustrant la convergence vers le 30<sup>e</sup> epoch.

Comme illustré dans la figure 6.6, l'augmentation des valeurs de  $n$  dans les métriques NDCG (passant de @10 à @100) entraîne une augmentation générale des scores, ce qui est attendu puisque des métriques plus larges sont moins strictes. Cependant, la réelle amélioration provient de l'affinage du modèle : nous observons des gains marqués après seulement quatre epochs, en particulier pour les embeddings de plus petite dimension, où l'impact de l'ajustement est le plus prononcé, mais les performances restent inférieures à celles obtenues avec un plus grand nombre de dimensions.

### 6.3.2 Récupération entre produits similaires

#### 6.3.2.1 Composition du jeu de données

Dans cette section, nous abordons la tâche de recherche de produits similaires. Concrètement, étant donné le contenu d'un produit (titre, description ou image), il s'agit d'identifier d'autres articles proches. Cette étape poursuit deux objectifs. D'une part, lorsqu'un utilisateur fournit l'image ou le titre d'un produit et souhaite trouver des articles analogues, nous pouvons exploiter les représentations textuelles ou visuelles issues d'un modèle d'embedding pour effectuer la recherche. D'autre part, comme nous l'avons mentionné dans la section 5.3.2, les énoncés des utilisateurs sont généralement formulés sous forme de dialogue spontané et

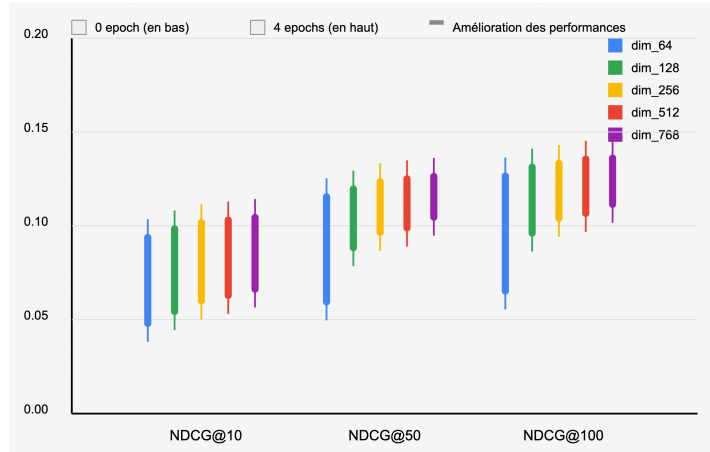


FIG. 6.6 : Comparaison des scores NDCG@10, 50 et 100 avant (epoch 0) et après fine-tuning (epoch 4) selon différentes dimensions d'embedding.

naturel, ce qui crée un écart intrinsèque avec les titres de produits. Cet écart est confirmé dans la section 6.3.1, où les scores de NDCG@100 restent faibles (systématiquement inférieur à 0,2%). Utiliser directement l'embedding d'un énoncé pour comparer sa similarité avec les embeddings des titres de produits conduit donc à un taux de succès limité, car les modèles d'embedding se contentent de projeter les textes en représentations vectorielles, sans disposer de la capacité de compréhension et de génération propre aux *LLMs*. C'est pourquoi nous avons introduit, dans la section 5.3.2, un module de générateur du contenu de produit comme une stratégie du système. Celui-ci permet au système, après avoir interprété l'intention de l'utilisateur au moyen d'un *LLM*, de générer un contenu de produit plus adapté au scénario commercial (par exemple un titre ou une description de produit). L'embedding de ce contenu synthétisé est ensuite utilisé pour rechercher des produits similaires. Dans la présente section, nous cherchons donc à évaluer si l'utilisation de contenus produits de cette manière améliore effectivement la précision de la recherche.

Nous comparons deux modèles existants, OpenAI CLIP ViT B/16 (RADFORD et al. 2021) et Jina-CLIP-v1 (XIAO, MASTRAPAS et B. WANG 2024). Dans l'article présentant ce dernier, les auteurs fournissent les scores de ces deux modèles, ainsi que d'autres, sur plusieurs benchmarks. Étant donné leur simplicité, leur légèreté et leurs bonnes performances, ces modèles ont été retenus comme candidats pour l'extraction des embeddings d'informations multimodales dans

notre système.

Afin d’adapter l’évaluation à notre scénario, nous utilisons le jeu de données Fashion200k (Xintong HAN et al. 2017), un corpus de grande échelle spécifiquement conçu pour les tâches de recherche d’images dans le domaine de la mode et de compréhension intermodale. Il contient plus de 200 000 images de produits de mode (vêtements, chaussures et accessoires), accompagnées de descriptions textuelles et de métadonnées associées.

Afin d’évaluer les performances des deux modèles (CLIP d’OpenAI et Jina-CLIP-v1), nous avons d’abord indexé l’ensemble du jeu de données. Chaque produit a été catégorisé à deux niveaux : une catégorie large (par exemple, robe) et une catégorie fine (par exemple, robe midi en maille), sur la base des descriptions textuelles et des métadonnées fournies dans le corpus.

Nous avons ensuite généré 1 000 requêtes textuelles pour l’évaluation, chacune étant construite à partir d’une catégorie large et d’une catégorie fine. Ces étiquettes associées nous permettent de quantifier la précision des modèles en mesurant le degré de correspondance entre les catégories des produits retournés et celles attendues à partir des requêtes initiales.

6.3.2.2 Comparaison des modèles

Nous avons d’abord comparé les performances en zero-shot des deux modèles sur les benchmarks CLIP<sup>2</sup> (CHERTI et BEAUMONT 2025). Nous rapportons les résultats de ces évaluations dans le tableau 6.12.

Recherche d'image			Recherche de texte		
Dataset	CLIP	Jina-CLIP	Dataset	CLIP	Jina-CLIP
Flickr30k	85.60	89.02	Flickr30k	96.20	96.50
Flickr8k	82.84	85.50	Flickr8k	91.40	94.20
MSCOCO	58.42	66.42	MSCOCO	76.76	79.02

TAB. 6.12 : Performances des modèles OpenAI CLIP ViT B/16 et Jina-CLIP-v1 sur les benchmarks CLIP, en recherche zéro-shot (Recall@5, en %)

On peut constater que les deux modèles obtiennent des performances similaires

<sup>2</sup>Les benchmarks CLIP regroupent un ensemble de tâches d’évaluation visuelle et textuelle (par ex. classification d’images, recherche image–texte, etc.) conçues pour mesurer la qualité des représentations produites par les modèles de type CLIP.



sur les benchmarks CLIP. Toutefois, ces benchmarks portent sur des données issues de domaines généraux. Nous poursuivons donc l'évaluation en comparant les deux modèles sur notre propre jeu de données, Fashion200K, spécifiquement conçu pour le domaine de la mode.

Ces deux modèles produisent des représentations textuelles et visuelles dans un même espace sémantique, avec des dimensions compatibles. Cela permet de comparer directement les embeddings texte et image, ou de les exploiter conjointement. Nous avons ainsi défini trois modes de recherche, chacun basé sur une même requête textuelle :

- Recherche requête-description : comparaison entre l'embedding texte de la requête et les embeddings texte des descriptions produits.
- Recherche requête-image : comparaison entre l'embedding texte de la requête et les embeddings image des produits.
- Recherche requête-moyenne (texte + image) : comparaison entre l'embedding texte de la requête et la moyenne des embeddings texte et image de chaque produit.

Nous ordonnons les produits en fonction de la similarité cosinus entre l'embedding de la requête et ceux des produits. Ensuite, nous calculons la précision@k en mesurant combien d'articles retournés appartenaient à la même catégorie (large ou fine) que celle associée à la requête parmi les premiers  $k$  résultats. Les scores sont présentés dans le tableau 6.13.

Nous observons que les deux modèles présentent des performances relativement satisfaisantes en récupération intermodale pour la requête vers l'image, avec des écarts très faibles. En revanche, Jina-CLIP surpasse clairement CLIP dans le cas de la requête vers l'embedding moyen. En nous appuyant également sur les résultats publiés dans des benchmarks de référence en domaine général pour CLIP (CHERTI et BEAUMONT 2025), nous estimons que CLIP reste une option plus légère et suffisante lorsque le système ne nécessite que la recherche d'image par image. En revanche, pour des cas d'usage impliquant la recherche de produit à partir du nom ou une combinaison texte-image, Jina-CLIP, plus récent, constitue un choix plus pertinent.

TAB. 6.13 : Précision@K pour les catégories larges et fines

K	Type	Catégories larges		Catégories fines	
		CLIP	Jina-CLIP	CLIP	Jina-CLIP
10	Query to Description	0.882	0.901	0.647	0.667
	Query to Image	0.736	0.738	0.570	0.596
	Query to Avg Embedding	0.909	0.924	0.664	0.690
20	Query to Description	0.893	0.912	0.660	0.678
	Query to Image	0.724	0.720	0.568	0.593
	Query to Avg Embedding	0.918	0.929	0.669	0.693
50	Query to Description	0.900	0.915	0.667	0.682
	Query to Image	0.702	0.708	0.562	0.586
	Query to Avg Embedding	0.921	0.931	0.674	0.696
100	Query to Description	0.899	0.912	0.666	0.679
	Query to Image	0.690	0.700	0.558	0.582
	Query to Avg Embedding	0.913	0.926	0.678	0.691

## 6.4 Simulateur du client

### 6.4.0.1 Composition du jeu de données

Nous avons d’abord extrait des données de commerce électronique telles que les descriptions de produits, les images, les avis et les chiffres de vente à partir des jeux de données de commerce électronique suivants. Nous avons utilisé plusieurs sources de données (à la fois du domaine e-commerce et du domaine général) et les exemples ont été transformés en consignes de style instruction :

#### Jeux de données du domaine e-commercial

- Amazon-M2 (W. JIN et al. 2024) : Un jeu de données complet des sessions utilisateurs sur Amazon, contenant les clics historiques ainsi que le clic actuel, enrichi de métadonnées détaillées sur les produits.
- Amazon Reviews 2023 (HOU et al. 2024) : Plus de 500 millions d’avis utilisateurs sur Amazon, accompagnés de vérifications d’achats, couvrant 33 catégories de produits.
- ECInstruct (BO PENG et al. 2024) : 116 528 échantillons dérivés de 10 tâches de commerce électronique à travers quatre catégories.

- ESCI-data (REDDY et al. 2022) : paires de requêtes d’achat et de résultats pertinents, complétées par des jugements de pertinence ESCI (Exact, Substitut, Complément, Non pertinent) pour évaluer la pertinence des produits.

#### Jeux de données du domaine général

- MMLU (HENDRYCKS et al. 2020) : un jeu de données multitâche massif composé de plus de 100 000 questions à choix multiples et de leurs réponses.
- Alpaca-Cleaned (TAORI et al. 2023) : une version nettoyée du jeu de données Alpaca original publié par l’université de Stanford.

De plus, nous avons identifié un chevauchement partiel entre les produits présents dans les différents jeux de données e-commerce d’Amazon. Nous avons fait une correspondance entre des attributs tels que les « clics historiques » provenant d’Amazon-M2 et les « requêtes » issues d’ESCI pour représenter la *requête utilisateur*, et utilisé des attributs comme les « achats vérifiés » du jeu de données Amazon Reviews 2023 ainsi que les « jugements de pertinence » d’ESCI comme étiquettes représentant l’*action utilisateur*. Nous avons ensuite construit un jeu de données tabulaire joint à partir de la colonne « produit actuel », aboutissant à un total de 2,6 millions de lignes (voir le tableau 6.14). Le tableau récapitule les colonnes principales : « requête utilisateur », « produit actuel », « contenu produit », et « action utilisateur ». Ce jeu de données fusionné a servi à entraîner notre modèle d’alignement du comportement utilisateur, nommé *Oracle*, pour simuler la décision d’achat du client.

TAB. 6.14 : Aperçu de la répartition du jeu de données des actions utilisateur

Répartition	Entraînement	Développement	Test	Total
<b>Nombre des instances</b>	2,080,000	260,000	260,000	2,600,000
<b>Proportion</b>	90%	10%	10%	100%

##### 6.4.0.2 Entraînement du simulateur « Oracle »

Nous avons adapté Amazon-M2 et Amazon Reviews 2023 afin que ces jeux de données soient aussi proches que possibles des caractéristiques du comportement utilisateur illustrés dans la figure 5.4, tels que les préférences de produits, les parcours de clics et les retours implicites. Les ensembles de données ECInstruct et

Modèle	Score F1	avec données synthétiques
Qwen1.5-14B	0,2003	0,2111
LlaMa3-70B-AWQ	0,6406	0,6771
ChatGPT-4o	0,8131	-
Smaug-72B-ZS	0,6564	0,6684
Smaug-72B-FT	0,6717	0,6907
Qwen2-72B-ZS	0,7783	0,7881
Qwen2-72B-FT	0,8113	<b>0,8235</b>

TAB. 6.15 : Comparaison de différents *LLMs* en zéro-shot (ZS) avec fine-tuning (FT) sur la prédiction de la décision du client.

ESCI-data ont fourni des tâches de commerce électronique structurées, permettant au modèle d'apprendre la catégorisation des produits et la prédiction de pertinence. Les ensembles de données MMLU et Alpaca-Cleaned ont offert une base plus large pour le raisonnement général et la génération de texte, respectivement. Le jeu de données fusionné décrit précédemment a été utilisé pour entraîner la capacité de prise de décision du modèle dans des contextes d'achat. Nous utilisons également GPT-4o-2024-05-13 (OPENAI 2023) pour générer des données synthétiques pour les domaines sous-représentés. Ces exemples synthétiques ont introduit de la diversité dans l'ensemble d'entraînement, aidant le modèle à apprendre à gérer un éventail plus large de comportements et d'interactions utilisateur. Cette approche globale a permis au modèle d'atteindre des performances élevées dans un large éventail de tâches d'alignement du comportement utilisateur (DEOTTE et al. 2024). Nous avons ensuite conçu des instructions qui guident le modèle pendant l'ajustement fin, les détails de la conception des instructions sont présentés dans l'annexe A.5.

Nous avons évalué la performance zero-shot de qwen1.5-14B (QWEN TEAM 2024), LlaMa3-70B-AWQ<sup>3</sup>, Smaug-72B (PAL et al. 2024) et Qwen2-72B-Instruct (A. YANG et al. 2024) et ChatGPT-4o-2024-05-13 (OPENAI 2023) sur les benchmarks. Les résultats expérimentaux sont présentés dans le tableau 6.15. Les F-scores observés sont globalement satisfaisants, ce qui montre que le simulateur utilisateur est capable de reproduire de manière crédible les décisions d'un client dans nos dialogues. De plus, l'ajustement fin sur des données synthétiques améliore

<sup>3</sup><https://huggingface.co/TechxGenus/Meta-Llama-3-70B-AWQ>

systématiquement les performances, confirmant l'intérêt de cette approche pour affiner le comportement du modèle. Le modèle **Qwen2-72B-FT**, en montrant un taux de confiance de précision de 82% lorsque l'on utilise les données synthétique, a été considéré comme le meilleur modèle pour servir de notre modèle **Oracle**.

## 6.5 Optimisation Kahneman-Tversky

### 6.5.1 Configuration expérimentale

Comme nous avons introduit dans la section 5.4, l'objectif de cette optimisation est d'entraîner notre modèle à générer des pseudo-descriptions de produits en se basant sur une compréhension adéquate du besoin client, de manière à ce que le simulateur client Oracle les considère comme satisfaisantes et prenne la décision d'acheter. Pour ce faire, nous utilisons le jeu de données fusionné (voir la section 6.4.0.1) à des fins d'entraînement et de test. La colonne "requête" est utilisée comme entrée, la colonne "description du produit" comme référence, et la décision simulée par Oracle sert de fonction de valeur de *KTO*.

Pour toutes les expériences, nous avons utilisé notre jeu de données d'entraînement avec 8 GPU NVIDIA A100 avec 80 Go de mémoire GPU. L'entraînement sur 500 000 exemples prend environ 24 heures. Pour utiliser pleinement le matériel, nous avons utilisé Colossal-AI (Shenggui LI et al. 2023) avec quantification 4 bits et bfloat16 (BURGESS et al. 2019). Nous avons affiné les modèles avec des hyper-paramètres identiques. Nous avons utilisé l'optimiseur AdamW avec un planificateur de taux d'apprentissage en cosinus sur un taux d'apprentissage de 0,0002 avec une décroissance de poids de 0,01. Nous avons défini les étapes de préchauffage à 10 et la taille du lot à 1 pour chaque GPU.

Pour les hyper-paramètres dans les équations 5.18 et 5.4.2 (voir la section 5.4), nous avons suivi la pratique proposée par Z. CHENG et al. (2025). Nous rappelons que  $\alpha$  et  $\lambda$  reflètent la façon réaliste dont les comportements humains s'adaptent face aux pertes et aux gains. Une forte aversion aux pertes correspond à des valeurs de  $\lambda$  significativement supérieures à 1, tandis que  $\alpha$  peut varier en fonction du contexte décisionnel particulier. D'après le travail mentionné (Z. CHENG et al. 2025), les valeurs médianes des hyper-paramètres sont  $\alpha = 0.88$  et  $\lambda = 2.25$ , qui correspondent à des estimations représentatives observées empiriquement chez différents individus. Ces paramètres influencent directement

Catégorie	#Articles	CVR de base	Qwen-ZS CVR	Qwen-FT CVR	ChatGPT-4o CVR	Qwen-KTSA CVR
Toute beauté	112,6K	2,11	2,08	2,08	2,13	2,17
Appareils électroménagers	94,3K	1,13	1,13	1,13	1,10	1,30
Produits pour bébés	217,7K	2,71	2,66	2,53	2,16	2,78
Beauté et soins	1,0M	2,11	1,90	1,91	2,01	2,13
Livres	4,4M	1,93	1,90	1,91	1,90	1,96
Apparence	7,2M	1,95	1,90	1,93	2,11	2,05
Fournitures pour animaux	492,7K	2,14	1,93	2,10	2,19	2,19
Jouets et jeux	890,7K	2,85	2,55	2,53	2,51	2,82

TAB. 6.16 : Le taux de conversion moyen (CVR) dans le commerce électronique avec différentes configurations de LLM pour différentes catégories. Nous avons suivi les catégories définies dans Amazon Reviews 2023. Tous les CVR sont mesurés en pourcentage. Le taux de conversion de base est mesuré et agrégé par le modèle Oracle (Qwen-2 72B-FT). Les nombres marqués en **rouge** sont inférieurs au CVR de base. Le meilleur CVR est indiqué en caractères **gras**.

la manière dont notre modèle simule les processus de prise de décision et les préférences des consommateurs face aux produits recommandés. Quant au paramètre  $\gamma$ , qui est un facteur de régularisation représentant le poids entre la valeur « acheteur » et la valeur « vendeur », nous l’avons fixé à  $\gamma = 1$  dans tous les cas.

### 6.5.2 Résultats expérimentaux

Nous générons des descriptions de produits par plusieurs LLMs et simulons les actions des consommateurs avec notre précédent modèle Oracle (Qwen-2 72B-FT). Nous évaluons le taux de conversion moyen (CVR) pour 8 catégories différentes pour chaque modèle. Dans les contextes de commerce électronique, plus le CVR est élevé, mieux le modèle s’aligne sur les préférences des consommateurs et facilite des recommandations de produits efficaces. Le CVR moyen du commerce électronique se situe entre 1% et 4%, mais peut varier considérablement selon l’industrie, le modèle commercial et d’autres facteurs.

Nous suivons les pratiques de ETHAYARAJH et al. 2024 et fixons  $\lambda_D = \lambda_U = 1$  dans l’équation 5.20 pour entraîner notre modèle Qwen-KTSA. Par rapport au CVR de base, dans la catégorie ”Appareils électroménagers”, nous avons obtenu une amélioration de 15%, passant de 1,13 à 1,30. Dans la catégorie ”Fournitures pour animaux”, nous avons obtenu les mêmes performances que ChatGPT-4o.

Cependant, dans la catégorie "Apparence", bien que nous ayons obtenu une amélioration de 5%, nous restons inférieurs à ChatGPT-4o. Globalement, les résultats montrent que les contenus générés par Qwen-KTSA surpassent ceux des autres modèles dans 7 des 8 catégories, démontrant sa capacité à comprendre les besoins latents des utilisateurs et à générer des contenus précis afin d'effectuer des sélections de produits appropriées.





Quatrième partie

Perspectives



## Chapitre 7

---

### Discussion et conclusion

Les travaux décrits dans cette thèse ont porté sur le développement d'un système conversationnel personnalisé et multimodal, basé sur des modèles de langage de grande taille (*LLM*), spécifiquement conçu pour le domaine du commerce électronique dans le secteur de la mode. L'architecture adoptée repose sur une structure en chaîne de traitement.

Pour ce faire, nous avons organisé le système à l'aide de deux éléments principaux : données et modèles. D'une part, nous avons collecté et généré un volume important de données, incluant des dialogues et des bases de connaissances. Nous avons également adapté et réorganisé des jeux de données existants afin de concevoir des tâches correspondant précisément à nos objectifs. D'autre part, nous avons comparé et adapté plusieurs modèles de base afin de sélectionner et d'intégrer les plus performants au sein de la chaîne de traitement.

Enfin, pour l'entraînement du système complet, nous avons mis en œuvre une stratégie d'auto-alignement (*self-alignment*) faisant intervenir un simulateur d'utilisateur. Ce dernier modélise les préférences et comportements des clients afin d'inciter le système à produire des contenus plus pertinents et à améliorer la qualité des recommandations.

#### 7.1 Discussion

Nos travaux ont commencé en 2019, à une période où l'utilisation des modèles d'apprentissage profond pour les systèmes de dialogue émergeaient tout juste. Les modèles visuels maîtrisaient déjà des tâches telles que la détection et la segmentation d'objets, tandis que les modèles linguistiques et visuels évoluaient progressivement des tâches de classification vers des tâches de génération. À partir de 2021, les approches multimodales ont suscité un intérêt croissant,

marquant un glissement général de la recherche — d’architectures spécialisées pour des tâches uniques vers des architectures plus complexes capables de traiter des tâches multiples et intégrées.

Cette tendance s’est reflétée de manière particulièrement visible dans le domaine des systèmes de dialogue. Les architectures traditionnelles séparaient le traitement de l’information en quatre modules distincts : *Natural Language Understanding*, (*NLU*), *Dialogue State Tracking*, (*DST*), *Policy Learning*, (*PL*) et *Natural Language Generation*, (*NLG*). Les travaux de l’époque commençaient à explorer des modèles de type de bout en bout (*end-to-end*) capables de prendre en charge l’ensemble de la chaîne de traitement. Progressivement, l’ensemble de la communauté TAL a concentré son attention sur ces nouvelles architectures, matérialisées par les grands modèles de langages (*LLMs*).

C’est donc naturellement que nos recherches se sont initialement orientées vers une approche « tout-en-un » reposant sur un modèle *end-to-end*. Cependant, nos expérimentations ont rapidement montré que cette approche ne permettait pas de gérer efficacement des volumes massifs d’informations produits, comme c’est le cas dans l’e-commerce de mode. Même en utilisant un *LLM* de très grande taille, lui faire « mémoriser » l’intégralité du catalogue produit s’est révélé inefficace et peu pertinent.

Nous avons donc réorienté notre conception vers une architecture hybride : conservant la modularité héritée des systèmes traditionnels, mais en isolant spécifiquement le module *DST* pour le traitement de la recherche et du classement des produits. Les autres modules — *NLU*, *PL* et *NLG* — ont été confiés à un *LLM* unique. Grâce à la polyvalence des *LLMs*, capables à la fois de comprendre et de générer du texte, nous avons pu réduire la séparation entre ces modules.

Concrètement, alors que les approches traditionnelles exigent du *NLU* qu’il identifie précisément l’intention de l’utilisateur et les attributs produits d’intérêt sous forme de paires *slot-value* à transmettre au *DST*, notre *NLU* n’a qu’à déterminer si l’entrée courante de l’utilisateur nécessite ou non une nouvelle recherche produit. En cas de besoin, le *DST* est invoqué pour extraire un nombre limité de produits dont les informations sont injectées dans le *prompt* qui sera passé au *LLM*. Une version plus avancée de cette méthode consiste à générer une description produit à partir de l’entrée utilisateur, puis à rechercher dans la base réelle les produits les plus proches en utilisant les embeddings correspondants. Si

aucune recherche n'est nécessaire, l'historique du dialogue est directement utilisé comme *prompt*.

Cette approche présente plusieurs avantages :

1. le *LLM* est entraîné et évalué indépendamment des données produits brutes ; il n'a pas à gérer simultanément mémoire, compréhension, comparaison et recherche, ce qui évite de recourir à un modèle de taille excessive ;
2. le modèle d'embedding peut être spécifiquement optimisé pour comparer la similarité produit-produit, plutôt que requête-produit ;
3. dans un contexte multimodal, la comparaison reste également centrée sur produit-produit (image et texte), plutôt que sur l'appariement complexe requête-produit.

Enfin, nous avons affiné le même *LLM* que celui employé pour l'agent système, afin qu'il joue également le rôle de simulateur utilisateur : à simuler la prise de décision d'un acheteur et à déterminer si celui-ci est susceptible d'apprécier les recommandations du système. Cette approche nous a permis, à coût réduit, de mettre en place un mécanisme de rétroaction optimisant le modèle agent, tout en offrant un protocole d'évaluation économique et pertinent.

## 7.2 Limitations et perspectives

Notre travail, bien qu'il propose une avancée dans la conception de systèmes de dialogue personnalisés et multimodaux pour l'e-commerce de mode, présente plusieurs limites et ouvre diverses pistes de recherche futures. Nous les discutons ci-dessous.

**1. Contraintes computationnelles et énergétiques.** Le recours à des *LLMs* soulève inévitablement des enjeux en termes de ressources. L'entraînement et l'inférence nécessitent l'utilisation de GPU coûteux et énergivores. Dans un contexte où les lois d'échelle (*scale laws*) poussent à utiliser des modèles toujours plus volumineux, la communauté se trouve engagée dans une véritable course à la puissance de calcul brute. Or, cette tendance soulève des préoccupations économiques et écologiques : à court terme, il est probable que l'utilisation intensive de modèles massifs reste dominante ; à long terme, il deviendra crucial de concevoir des architectures plus sobres et durables, afin de concilier performance, efficacité énergétique et responsabilité environnementale.

**2. Données, générativité et qualité.** La disponibilité et la qualité des données constituent une seconde limite. Selon les mêmes lois d'échelle, des modèles plus grands requièrent des volumes de données considérables. Dans le chapitre 4.3.1, nous avons montré qu'il existe une sorte de « triangle d'impossibilité » en matière d'acquisition de données : il est difficile d'obtenir à la fois des données massives, fiables et peu coûteuses. L'utilisation de données synthétiques constitue une solution partielle, mais elle comporte le risque de produire des modèles qui ne font que « répéter » les biais de leurs modèles enseignants (*parroting effect*). Afin d'atténuer ce risque, nous avons fait appel à des annotateurs humains pour évaluer la qualité d'un sous-ensemble de dialogues synthétiques (chapitre 6.1.2), et nous avons veillé à utiliser des modèles enseignants fiables. Par ailleurs, pour entraîner le simulateur d'utilisateurs, nous avons transformé de grandes quantités de données réelles issues de l'e-commerce en comportements utilisateurs spécifiques à nos tâches cibles (section 6.4.0.2). Bien que cette approche conserve des traces d'artificialité, elle constitue un compromis efficace en termes de coût et de pertinence.

**3. Interaction et expérience utilisateur.** Du point de vue des utilisateurs, l'interaction conversationnelle offre plusieurs avantages par rapport aux méthodes traditionnelles de recherche ou de recommandation de produits : elle est plus naturelle, continue et procure un sentiment de « service personnalisé ». En effet, les utilisateurs ne sont pas des commerçants et ne maîtrisent pas toujours le langage de recherche précis correspondant aux titres produits ; il existe ainsi un écart structurel entre le langage courant et la terminologie commerciale. Les systèmes de recherche classiques, notamment via le filtrage collaboratif (*collaborative filtering*), compensent partiellement ce décalage. Néanmoins, certaines situations soulignent encore la pertinence des approches traditionnelles : par exemple, lorsqu'un utilisateur n'a pas d'intention d'achat ferme et souhaite simplement « parcourir » une offre, les interfaces de recherche et de navigation classiques peuvent mieux répondre à ce besoin, tout en favorisant parfois des achats impulsifs. Ces considérations suggèrent que les systèmes conversationnels ne doivent pas être envisagés comme un remplacement exclusif, mais plutôt comme un complément intégré dans une expérience utilisateur hybride.

4. **Enjeux éthiques et sociétaux.** Enfin, notre travail s’inscrit dans un cadre éthique plus large. Plusieurs dimensions méritent d’être discutées :

**Impact environnemental** : comme mentionné plus haut, l’empreinte carbone liée à l’entraînement et au déploiement de LLM massifs constitue une préoccupation majeure.

**Automatisation et emploi** : l’automatisation croissante des tâches de conseil et d’assistance soulève des craintes légitimes concernant la substitution de la main-d’œuvre humaine par des systèmes automatisés.

**Éthique des données** : l’utilisation de données massives, qu’elles soient réelles ou synthétiques, implique des enjeux de confidentialité, de biais et de transparence. Une vigilance accrue est nécessaire pour garantir que les modèles ne reproduisent pas ou n’amplifient pas des inégalités existantes.

**Éthique spécifique à l’e-commerce** : un aspect souvent négligé concerne ce que l’on pourrait qualifier de « fiscalité rose » (*pink tax*). Initialement observée dans des différences de prix entre produits genrés mais identiques (par exemple, des rasoirs féminins plus chers que leurs équivalents masculins), cette logique se retrouve aujourd’hui dans la manipulation discursive des titres et descriptions produits. Des formulations telles que « cardigan rétro style MiuMiu » ou « panier artisanal préféré des chats » peuvent correspondre au même produit que de simples intitulés comme « cardigan gris pour homme » ou « panier en paille pour volailles ». Ces pratiques, bien qu’elles puissent accroître les revenus à court terme, posent un problème d’équité et de transparence : elles incitent les consommateurs à payer davantage pour une valeur ajoutée essentiellement discursive, tout en décourageant les commerçants honnêtes et en fragilisant la confiance envers les plateformes à long terme.

## 7.3 Conclusion générale

Cette thèse a exploré la conception et la mise en œuvre d’un système de dialogue personnalisé et multimodal pour l’e-commerce de mode, reposant sur des modèles de langage de grande taille et sur une architecture en chaîne de traitement. Notre démarche s’est articulée autour de deux axes principaux : d’une part, la constitution et la génération de données adaptées, incluant des dialogues, des bases de connaissances et des annotations de produits, et d’autre part, l’exploration, la comparaison et l’intégration de différents modèles de base

afin d’aboutir à un système complet et cohérent.

Sur le plan méthodologique, nous avons montré qu’une approche purement *end-to-end*, bien que séduisante en théorie, se heurte à des limites pratiques liées à la gestion de vastes inventaires produits. En réponse, nous avons proposé une architecture hybride : un module de suivi d’état (*DST*) spécifiquement dédié à la recherche et au classement de produits, couplé à un modèle de langage capable d’assurer les autres fonctions du dialogue (*NLU*, *PL*, *NLG*) de manière intégrée. Cette approche a permis de combiner la modularité et l’efficacité de l’ancienne génération de systèmes avec la souplesse et la puissance des modèles de langage contemporains.

Sur le plan de l’innovation, notre travail a introduit un mécanisme de *self-alignment*, reposant sur un simulateur d’utilisateurs capable de modéliser des préférences et des comportements d’achat. Ce dispositif a servi à la fois à entraîner le système de manière plus économique et à proposer une méthodologie d’évaluation alternative. De plus, nous avons exploré l’utilisation de données synthétiques comme levier d’extension, tout en mettant en place des garde-fous (évaluations humaines, sélection rigoureuse des modèles enseignants) afin d’en assurer la qualité.

Les contributions de cette recherche sont doubles. D’un point de vue académique, elle met en évidence une voie médiane entre l’approche *end-to-end* et l’architecture modulaire traditionnelle, en tirant parti des capacités génératives des LLM tout en préservant la robustesse d’un module spécialisé pour la gestion de l’inventaire, et en proposant une solution globalement moins coûteuse en ressources. D’un point de vue applicatif, elle ouvre la perspective de systèmes de dialogue capables d’offrir une expérience utilisateur plus naturelle et personnalisée, adaptée aux contraintes spécifiques de l’e-commerce de mode.

Enfin, les discussions menées sur les limites de notre approche rappellent l’importance des enjeux computationnels, énergétiques, éthiques et sociétaux qui accompagnent l’essor des systèmes d’IA de grande échelle. À ce titre, plusieurs perspectives de recherche futures méritent d’être explorées : la conception de modèles plus sobres en ressources, le développement de méthodes de génération de données plus fiables et moins biaisées, l’intégration fluide des systèmes conversationnels et des interfaces de recherche traditionnelles, ainsi qu’une réflexion approfondie sur les pratiques commerciales et leur impact sur la confiance des



consommateurs.

En définitive, cette thèse entend apporter une contribution à la fois théorique et pratique à la construction de systèmes de dialogue multimodaux et personnalisés, tout en soulignant que l'avenir de tels systèmes ne se joue pas uniquement sur le plan technique, mais également sur le terrain des valeurs sociétales et de l'éthique de l'innovation.



# Bibliographie

- ACHIAM, OpenAI Josh et al. (2023). "GPT-4 Technical Report". In : URL : <https://api.semanticscholar.org/CorpusID:257532815>.
- AGARWAL, Shubham et al. (2020). "History for Visual Dialog : Do we really need it ?" In : *Annual Meeting of the Association for Computational Linguistics*.
- AGHAJANYAN, Armen et al. (nov. 2020). "Conversational Semantic Parsing". In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, p. 5026-5035. DOI : 10.18653/v1/2020.emnlp-main.408. URL : <https://aclanthology.org/2020.emnlp-main.408>.
- AGRAWAL, Aishwarya et al. (2015). "VQA : Visual Question Answering". In : *International Journal of Computer Vision* 123, p. 4-31.
- ALBERTI, Chris et al. (2019). "Fusion of Detected Objects in Text for Visual Question Answering". In : *ArXiv abs/1908.05054*.
- ARORA, Suket, Kamaljeet BATRA et Sarabjit SINGH (2013). "Dialogue System : A Brief Review". In : *ArXiv abs/1306.4134*.
- ASRI, Layla El et al. (2017). "Frames : a corpus for adding memory to goal-oriented dialogue systems". In : *ArXiv abs/1704.00057*.
- BAHDANAU, Dzmitry, Kyunghyun CHO et Yoshua BENGIO (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In : *CoRR abs/1409.0473*.
- BAHETI, Ashutosh, Alan RITTER et Kevin SMALL (2020). "Fluent Response Generation for Conversational Question Answering". In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:218763640>.
- BAI, Jinze et al. (2023). "Qwen Technical Report". In : *arXiv preprint arXiv :2309.16609*.
- BAIDOO-ANU, David et Leticia OWUSU ANSAH (2023). "Education in the era of generative artificial intelligence (AI) : Understanding the potential benefits

- of ChatGPT in promoting teaching and learning”. In : *Available at SSRN* 4337484.
- BALAKRISHNAN, Anusha et al. (2019). “Constrained Decoding for Neural NLG from Compositional Representations in Task-Oriented Dialogue”. In : *Annual Meeting of the Association for Computational Linguistics*.
- BALARAMAN, Vevake et Bernardo MAGNINI (2020). “Proactive systems and influenceable users : Simulating proactivity in task-oriented dialogues”. In : *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue-Full Papers, Virtually at Brandeis, Waltham, New Jersey, July. SEMDIAL*.
- BAPNA, Ankur et al. (2017). “Towards Zero-Shot Frame Semantic Parsing for Domain Scaling”. In : *Interspeech*. URL : <https://api.semanticscholar.org/CorpusID:4177915>.
- BORDES, Antoine et Jason WESTON (2016). “Learning End-to-End Goal-Oriented Dialog”. In : *ArXiv abs/1605.07683*.
- BOYD, Alex et al. (2020). “Large Scale Multi-Actor Generative Dialog Modeling”. In : *Annual Meeting of the Association for Computational Linguistics*.
- BROWN, Tom B. et al. (2020). “Language models are few-shot learners”. In : *Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20*. Vancouver, BC, Canada : Curran Associates Inc. ISBN : 9781713829546.
- BUDZIANOWSKI, Paweł et al. (2018). “MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling”. In : *Conference on Empirical Methods in Natural Language Processing*.
- BURGESS, Neil et al. (2019). “Bfloat16 processing for neural networks”. In : *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*. IEEE, p. 88-91.
- BURKE, Robin D, Kristian J HAMMOND et BC YOUNG (1997). “The FindMe approach to assisted browsing”. In : *IEEE Expert* 12.4, p. 32-40.
- BYRNE, Bill et al. (2019). *Taskmaster-1 : Toward a Realistic and Diverse Dialog Dataset*. arXiv : 1909.05358 [cs.CL].
- CAHILL, Lynne J. et al. (1999). “In search of a reference architecture for NLG systemsLynne”. In : URL : <https://api.semanticscholar.org/CorpusID:15051570>.

- CHEN, Haokun et al. (2019). "Large-scale interactive recommendation with tree-structured policy gradient". In : *Proceedings of the AAAI conference on artificial intelligence*. T. 33. 01, p. 3312-3320.
- CHEN, Hongshen et al. (2017). "A Survey on Dialogue Systems : Recent Advances and New Frontiers". In : *ArXiv abs/1711.01731*.
- CHEN, J. et al. (2020). "Parallel Interactive Networks for Multi-Domain Dialogue State Generation". In : *ArXiv abs/2009.07616*. URL : <https://api.semanticscholar.org/CorpusID:221738940>.
- CHEN, Li et Pearl PU (2012). "Critiquing-based recommenders : survey and emerging trends". In : *User Modeling and User-Adapted Interaction* 22, p. 125-150.
- CHEN, Meng et al. (2019). "The JDDC Corpus : A Large-Scale Multi-Turn Chinese Dialogue Dataset for E-commerce Customer Service". In : *ArXiv abs/1911.09969*.
- (2020). "The JDDC Corpus : A Large-Scale Multi-Turn Chinese Dialogue Dataset for E-commerce Customer Service". In : *ArXiv abs/1911.09969*.
- CHEN, Qibin et al. (2019). "Towards knowledge-based recommender dialog system". In : *arXiv preprint arXiv :1908.05391*.
- CHEN, Wenhui et al. (2019). "Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention". In : *Annual Meeting of the Association for Computational Linguistics*.
- CHEN, Xiuyi, Fandong MENG et al. (2020). "Bridging the Gap between Prior and Posterior Knowledge Selection for Knowledge-Grounded Dialogue Generation". In : *Conference on Empirical Methods in Natural Language Processing*.
- CHEN, Xiuyi, Jiaming XU et Bo XU (2019). "A Working Memory Model for Task-oriented Dialog Response Generation". In : *Annual Meeting of the Association for Computational Linguistics*.
- CHEN, Yen-Chun et al. (2019). "UNITER : Learning UNiversal Image-TExt Representations". In : *ArXiv abs/1909.11740*.
- CHENG, Jianpeng et al. (2020). "Conversational Semantic Parsing for Dialog State Tracking". In : *ArXiv abs/2010.12770*.
- CHENG, Zehua et al. (jan. 2025). "On Weaponization-Resistant Large Language Models with Prospect Theoretic Alignment". In : *Proceedings of the 31st International Conference on Computational Linguistics*. Sous la dir. d'Owen

- RAMBOW et al. Abu Dhabi, UAE : Association for Computational Linguistics, p. 10309-10324. URL : <https://aclanthology.org/2025.coling-main.687/>.
- CHERTI, Mehdi et Romain BEAUMONT (mai 2025). *CLIP benchmark*. DOI : 10.5281/zenodo.15403103. URL : <https://doi.org/10.5281/zenodo.15403103>.
- CHIU, Ssu et al. (2022). “SalesBot : Transitioning from Chit-Chat to Task-Oriented Dialogues”. In : *ArXiv abs/2204.10591*.
- CHO, Kyunghyun et al. (2014). “On the Properties of Neural Machine Translation : Encoder-Decoder Approaches”. In : *SSST@EMNLP*.
- CHOWDHERY, Aakanksha et al. (2022). “PaLM : Scaling Language Modeling with Pathways”. In : *ArXiv abs/2204.02311*. URL : <https://api.semanticscholar.org/CorpusID:247951931>.
- CHRISTAKOPOULOU, Konstantina, Alex BEUTEL et al. (2018). “Q&R : A two-stage approach toward interactive recommendation”. In : *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 139-148.
- CHRISTAKOPOULOU, Konstantina, Filip RADLINSKI et Katja HOFMANN (2016). “Towards conversational recommender systems”. In : *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, p. 815-824.
- COGSWELL, Michael et al. (2020). “Dialog without Dialog Data : Learning Visual Dialog Agents from VQA Data”. In : *ArXiv abs/2007.12750*.
- COOPE, Sam et al. (2020). “Span-ConveRT : Few-shot Span Extraction for Dialog with Pretrained Conversational Representations”. In : *ArXiv abs/2005.08866*.
- CROOK, Paul A et al. (2019). “SIMMC : Situated Interactive Multi-Modal Conversational Data Collection And Evaluation Platform”. In : *arXiv preprint arXiv :1911.02690*.
- DAI, Yinpei et al. (2020). “Learning Low-Resource End-To-End Goal-Oriented Dialog for Fast and Reliable System Deployment”. In : *Annual Meeting of the Association for Computational Linguistics*.
- DAS, Abhishek et al. (2016). “Visual Dialog”. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1080-1089.
- DE VIN, Leo J. et al. (2006). “Information fusion for simulation based decision support in manufacturing”. In : *Robotics and Computer-Integrated Manufactu-*

- ring 22.5. 15th International Conference on Flexible Automation and Intelligent Manufacturing, p. 429-436. ISSN : 0736-5845. DOI : <https://doi.org/10.1016/j.rcim.2005.11.007>. URL : <https://www.sciencedirect.com/science/article/pii/S0736584506000536>.
- DEOTTE, Chris et al. (2024). “Winning Amazon KDD Cup’24”. In : *ArXiv abs/2408.04658*. URL : <https://api.semanticscholar.org/CorpusID:271843298>.
- DETTMERS, Tim et al. (2023). “QLoRA : Efficient Finetuning of Quantized LLMs”. In : *ArXiv abs/2305.14314*. URL : <https://api.semanticscholar.org/CorpusID:258841328>.
- DHINGRA, Bhuwan et al. (2016). “Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:8951658>.
- DINAN, Emily et al. (2019). “The Second Conversational Intelligence Challenge (ConvAI2)”. In : *ArXiv abs/1902.00098*.
- DING, Ning et al. (2023). “Enhancing Chat Language Models by Scaling High-quality Instructional Conversations”. In : *ArXiv abs/2305.14233*. URL : <https://api.semanticscholar.org/CorpusID:258840897>.
- DU, Nan et al. (2019). “Extracting Symptoms and their Status from Clinical Conversations”. In : *Annual Meeting of the Association for Computational Linguistics*.
- DU, Wenchao et Alan W. BLACK (2019). “Boosting Dialog Response Generation”. In : *Annual Meeting of the Association for Computational Linguistics*.
- DUSEK, Ondrej et Filip JURČÍČEK (2016a). “A Context-aware Natural Language Generator for Dialogue Systems”. In : *ArXiv abs/1608.07076*. URL : <https://api.semanticscholar.org/CorpusID:3201741>.
- (2016b). “Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings”. In : *ArXiv abs/1606.05491*. URL : <https://api.semanticscholar.org/CorpusID:6380915>.
- ELDER, Henry, Alexander O’CONNOR et Jennifer FOSTER (2020). “How to Make Neural Natural Language Generation as Reliable as Templates in Task-Oriented Dialogue”. In : *Conference on Empirical Methods in Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:226262222>.

- ENEVOLDSEN, Kenneth et al. (2025). “MMTEB : Massive Multilingual Text Embedding Benchmark”. In : *arXiv preprint arXiv :2502.13595*. DOI : 10.48550/arXiv.2502.13595. URL : <https://arxiv.org/abs/2502.13595>.
- ERIC, Mihail et al. (2017). “Key-Value Retrieval Networks for Task-Oriented Dialogue”. In : *ArXiv abs/1705.05414*.
- ETHAYARAJH, Kawin et al. (2024). “KTO : Model alignment as prospect theoretic optimization”. In : *arXiv preprint arXiv :2402.01306*.
- EVANS, Owain et al. (2021). *Truthful AI : Developing and governing AI that does not lie*. arXiv : 2110.06674 [cs.CY].
- FELBER, HELMUT (1980). In : *International Journal of the Sociology of Language* 1980.23, p. 65-80. DOI : doi:10.1515/ijsl.1980.23.65. URL : <https://doi.org/10.1515/ijsl.1980.23.65>.
- FELDMAN, Yair et Ran EL-YANIV (2019). “Multi-Hop Paragraph Retrieval for Open-Domain Question Answering”. In : *ArXiv abs/1906.06606*.
- FENG, Jiazhan et al. (2022). *MMDialog : A Large-scale Multi-turn Dialogue Dataset Towards Multi-modal Open-domain Conversation*. arXiv : 2211.05719 [cs.CL].
- FERNANDEZ, Miriam et al. (jan. 2014). “User profile modelling in online communities”. In : *CEUR Workshop Proceedings* 1275, p. 35-48.
- GAN, Zhe, Yen-Chun CHEN et al. (2020). “Large-Scale Adversarial Training for Vision-and-Language Representation Learning”. In : *ArXiv abs/2006.06195*.
- GAN, Zhe, Yu CHENG et al. (2019). “Multi-step Reasoning via Recurrent Dual Attention for Visual Dialog”. In : *ArXiv abs/1902.00579*.
- GAO, Chongming et al. (2021). “Advances and Challenges in Conversational Recommender Systems : A Survey”. In : *AI Open* 2, p. 100-126. URL : <https://api.semanticscholar.org/CorpusID:231698518>.
- GAO, Silin et al. (2020). “Paraphrase Augmented Task-Oriented Dialog Generation”. In : *ArXiv abs/2004.07462*. URL : <https://api.semanticscholar.org/CorpusID:215786013>.
- GEMINI TEAM GOOGLE (2023). “Gemini : A Family of Highly Capable Multimodal Models”. In : *arXiv preprint arXiv :2312.11805*.
- GILOTTE, Alexandre et al. (2018). “Offline a/b testing for recommender systems”. In : *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, p. 198-206.



- GODDEAU, David et al. (1996). "A form-based dialogue manager for spoken language applications". In : *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96* 2, 701-704 vol.2. URL : <https://api.semanticscholar.org/CorpusID:13880916>.
- GOLOVANOV, Sergey et al. (2019). "Large-Scale Transfer Learning for Natural Language Generation". In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:196184953>.
- HA, Yu-I et al. (avr. 2017). "Fashion Conversation Data on Instagram". In : *Proceedings of the International AAAI Conference on Web and Social Media* 11. DOI : 10.1609/icwsm.v11i1.14858.
- HABER, Janosch et al. (2019). "The PhotoBook Dataset : Building Common Ground through Visually-Grounded Dialogue". In : *Annual Meeting of the Association for Computational Linguistics*.
- HAKKANI-TÜR, Dilek Z. et al. (2016). "Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM". In : *Interspeech*. URL : <https://api.semanticscholar.org/CorpusID:5775306>.
- HAM, Dong-hyun et al. (2020). "End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2". In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:219719687>.
- HAN, Xiao et al. (2022). *FashionViL : Fashion-Focused Vision-and-Language Representation Learning*. arXiv : 2207.08150 [cs.CV].
- HAN, Xintong et al. (2017). "Automatic Spatially-Aware Fashion Concept Discovery". In : *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 1472-1480. URL : <https://api.semanticscholar.org/CorpusID:3342857>.
- HASHEMI, Homa Baradaran (2016). "Query Intent Detection using Convolutional Neural Networks". In : URL : <https://api.semanticscholar.org/CorpusID:18908839>.
- HE, Chen, Denis PARRA et Katrien VERBERT (2016). "Interactive recommender systems : A survey of the state of the art and future research challenges and opportunities". In : *Expert Systems with Applications* 56, p. 9-27.
- HE, Xiangnan, Kuan DENG et al. (2020). "Lightgcn : Simplifying and powering graph convolution network for recommendation". In : *Proceedings of the*

- 43rd International ACM SIGIR conference on research and development in Information Retrieval, p. 639-648.
- HE, Xiangnan, Lizi LIAO et al. (2017). “Neural Collaborative Filtering”. In : *Proceedings of the 26th International Conference on World Wide Web*. URL : <https://api.semanticscholar.org/CorpusID:13907106>.
- HENDERSON, James, Oliver LEMON et Kallirroi GEORGILA (2008). “Hybrid Reinforcement/Supervised Learning of Dialogue Policies from Fixed Data Sets”. In : *Computational Linguistics* 34, p. 487-511. URL : <https://api.semanticscholar.org/CorpusID:9457948>.
- HENDERSON, Matthew, Blaise THOMSON et J. WILLIAMS (2014a). “The Second Dialog State Tracking Challenge”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:1294169>.
- (2014b). “The third Dialog State Tracking Challenge”. In : *2014 IEEE Spoken Language Technology Workshop (SLT)*, p. 324-329. URL : <https://api.semanticscholar.org/CorpusID:17478615>.
- HENDERSON, Matthew, Blaise THOMSON et Steve J. YOUNG (2013). “Deep Neural Network Approach for the Dialog State Tracking Challenge”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:2336451>.
- (2014a). “Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation”. In : *2014 IEEE Spoken Language Technology Workshop (SLT)*, p. 360-365. URL : <https://api.semanticscholar.org/CorpusID:2126884>.
- (2014b). “Word-Based Dialog State Tracking with Recurrent Neural Networks”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:8021351>.
- HENDERSON, Matthew, Ivan VULIC et al. (2019). “Training Neural Response Selection for Task-Oriented Dialogue Systems”. In : *ArXiv abs/1906.01543*.
- HENDRYCKS, Dan et al. (2020). “Measuring massive multitask language understanding”. In : *arXiv preprint arXiv :2009.03300*.
- HOSSEINI-ASL, Ehsan et al. (2020). “A Simple Language Model for Task-Oriented Dialogue”. In : *ArXiv abs/2005.00796*. URL : <https://api.semanticscholar.org/CorpusID:218487751>.
- HOU, Yupeng et al. (2024). “Bridging language and items for retrieval and recommendation”. In : *arXiv preprint arXiv :2403.03952*.

- HU, Jiaying et al. (2020). "SAS : Dialogue State Tracking via Slot Attention and Slot Information Sharing". In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:220046478>.
- HUANG, Jin et al. (2020). "Keeping dataset biases out of the simulation : A debiased simulator for reinforcement learning based recommender systems". In : *Proceedings of the 14th ACM Conference on Recommender Systems*, p. 190-199.
- HUANG, Y. et al. (2020). "Meta-Reinforced Multi-Domain State Generator for Dialogue Systems". In : *Annual Meeting of the Association for Computational Linguistics*.
- HUANG, Zhicheng et al. (2020). "Pixel-BERT : Aligning Image Pixels with Text by Deep Multi-Modal Transformers". In : *ArXiv abs/2004.00849*.
- J., Frances E. KELLEY et Thomas J. WATSON (1984). "An iterative design methodology for user-friendly natural language office information applications". In : *ACM Trans. Inf. Syst.* 2, p. 26-41. URL : <https://api.semanticscholar.org/CorpusID:207660078>.
- JAIN, Alankar et al. (2018). "A User Simulator Architecture for Socially-Aware Conversational Agents". In : *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. IVA '18. Sydney, NSW, Australia : Association for Computing Machinery, p. 133-140. ISBN : 9781450360135. DOI : 10.1145/3267851.3267916. URL : <https://doi.org/10.1145/3267851.3267916>.
- JI, Changzhen et al. (nov. 2020). "Cross Copy Network for Dialogue Generation". In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, p. 1900-1910. DOI : 10.18653/v1/2020.emnlp-main.149. URL : <https://aclanthology.org/2020.emnlp-main.149>.
- JIANG, Albert Q., Alexandre SABLAYROLLES, Arthur MENSCH et al. (2023). "Mistral 7B". In : *arXiv preprint arXiv :2310.06825*.
- JIANG, Albert Q., Alexandre SABLAYROLLES, Antoine ROUX et al. (2024). "Mixtral of Experts". In : *arXiv preprint arXiv :2401.04088*.
- JIN, Wei et al. (2024). "Amazon-M2 : A multilingual multi-locale shopping session dataset for recommendation and text generation". In : *Advances in Neural Information Processing Systems* 36.

- KALE, Mihir et Abhinav RASTOGI (2020). “Template Guided Text Generation for Task Oriented Dialogue”. In : *Conference on Empirical Methods in Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:226262225>.
- KIM, Hyunwoo, Byeongchang KIM et Gunhee KIM (2020). “Will I Sound like Me? Improving Persona Consistency in Dialogues through Pragmatic Self-Consciousness”. In : *Conference on Empirical Methods in Natural Language Processing*.
- KIM, Seokhwan et Rafael E. BANCHS (2014). “Sequential Labeling for Tracking Dynamic Dialog States”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:18632356>.
- KIM, Sungdong et al. (2019). “Efficient Dialogue State Tracking by Selectively Overwriting Memory”. In : *ArXiv abs/1911.03906*.
- KOSAR, Vaclav et al. (2022). “GLAMI-1M : A Multilingual Image-Text Fashion Dataset”. In : *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press. URL : <https://bmvc2022.mpi-inf.mpg.de/0607.pdf>.
- KOTTUR, Satwik, Seungwhan MOON et al. (2021). “Simmc 2.0 : A task-oriented dialog dataset for immersive multimodal conversations”. In : *arXiv preprint arXiv :2104.08667*.
- KOTTUR, Satwik, José M. F. MOURA et al. (2019). “CLEVR-Dialog : A Diagnostic Dataset for Multi-Round Reasoning in Visual Dialog”. In : *North American Chapter of the Association for Computational Linguistics*.
- KUNG, Tiffany H. et al. (2022). “Performance of ChatGPT on USMLE : Potential for AI-assisted medical education using large language models”. In : *PLOS Digital Health* 2.
- KUSUPATI, Aditya et al. (2022). “Matryoshka Representation Learning”. In : *Advances in Neural Information Processing Systems*.
- LARSON, Stefan et al. (2019). “An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction”. In : *ArXiv abs/1909.02027*.
- LE, Hung et al. (2020). “UniConv : A Unified Conversational Neural Architecture for Multi-domain Task-oriented Dialogues”. In : *ArXiv abs/2004.14307*. URL : <https://api.semanticscholar.org/CorpusID:216642138>.

- LEE, Harrison et al. (2022). "SGD-X : A Benchmark for Robust Generalization in Schema-Guided Dialogue Systems". In : 36.10, p. 10938-10946.
- LEE, Ji Young et Franck DERNONCOURT (2016). "Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks". In : *ArXiv* abs/1603.03827. URL : <https://api.semanticscholar.org/CorpusID:5591459>.
- LEE, Sungjin (2013). "Structured Discriminative Model For Dialog State Tracking". In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:1046547>.
- LEE, Sungjin et Maxine ESKÉNAZI (2013). "Recipe For Building Robust Spoken Dialog State Trackers : Dialog State Tracking Challenge System Description". In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:10079468>.
- LEE, Sungjin et Rahul JHA (2018). "Zero-Shot Adaptive Transfer for Conversational Language Understanding". In : *AAAI Conference on Artificial Intelligence*. URL : <https://api.semanticscholar.org/CorpusID:52135404>.
- LEE, Sungjin, Hannes SCHULZ et al. (2019). "Multi-Domain Task-Completion Dialog Challenge". In :
- LEE, Young-Jun, ByungSoo KO et al. (2022). "DialogCC : Large-Scale Multi-Modal Dialogue Dataset". In : *ArXiv* abs/2212.04119.
- LEE, Young-Jun, Chae-Gyun LIM et al. (2022). "PERSONACHATGEN : Generating Personalized Dialogues using GPT-3". In : *CCGPK*.
- LEI, Wenqiang, Xiangnan HE et al. (2020). "Estimation-action-reflection : Towards deep interaction between conversational and recommender systems". In : *Proceedings of the 13th International Conference on Web Search and Data Mining*, p. 304-312.
- LEI, Wenqiang, Xisen JIN et al. (2018). "Sequicity : Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures". In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:46927617>.
- LEI, Wenqiang, Gangyi ZHANG et al. (2020). "Interactive path reasoning on graph for conversational recommendation". In : *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, p. 2073-2083.

- LEWIS, Mike et al. (2017). "Deal or no deal ? end-to-end learning for negotiation dialogues". In : *arXiv preprint arXiv :1706.05125*.
- LEWIS, Patrick et al. (2020). "Retrieval-augmented generation for knowledge-intensive NLP tasks". In : *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada : Curran Associates Inc. ISBN : 9781713829546.
- LI, Gen et al. (2019). "Unicoder-VL : A Universal Encoder for Vision and Language by Cross-modal Pre-training". In : *AAAI Conference on Artificial Intelligence*.
- LI, Lihong, Jin Young KIM et Imed ZITOUNI (2015). "Toward predicting the outcome of an A/B experiment for search relevance". In : *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, p. 37-46.
- LI, Liunian Harold et al. (2019). "VisualBERT : A Simple and Performant Baseline for Vision and Language". In : *ArXiv abs/1908.03557*.
- LI, Raymond et al. (2018). "Towards deep conversational recommendations". In : *Advances in neural information processing systems* 31.
- LI, Shenggui et al. (2023). "Colossal-AI : A Unified Deep Learning System For Large-Scale Parallel Training". In : *Proceedings of the 52nd International Conference on Parallel Processing*. ICPP '23. Salt Lake City, UT, USA : Association for Computing Machinery, p. 766-775. ISBN : 9798400708435. DOI : 10.1145/3605573.3605613. URL : <https://doi.org/10.1145/3605573.3605613>.
- LI, Shijun et al. (2021). "Seamlessly unifying attributes and items : Conversational recommendation for cold-start users". In : *ACM Transactions on Information Systems (TOIS)* 39.4, p. 1-29.
- LI, Xiujun, Yun-Nung (Vivian) CHEN et al. (2017). "End-to-End Task-Completion Neural Dialogue Systems". In : *International Joint Conference on Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:18750779>.
- LI, Xiujun, Sarah PANDA et al. (2018). "Microsoft Dialogue Challenge : Building End-to-End Task-Completion Dialogue Systems". In : *ArXiv abs/1807.11125*.
- LI, Xiujun, Xi YIN et al. (2020). "Oscar : Object-Semantics Aligned Pre-training for Vision-Language Tasks". In : *European Conference on Computer Vision*.

- LI, Yangming et al. (2020). "Slot-consistent NLG for Task-oriented Dialogue Systems with Iterative Rectification Network". In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:220050117>.
- LIAO, Lizi et al. (2021). "MMConv : An Environment for Multimodal Conversational Search across Multiple Domains". In.
- LIAO, Yin-Hsiang et al. (mai 2022). "Conversational AI for Positive-sum Retailing under Falsehood Control". In : *Proceedings of the 4th Workshop on NLP for Conversational AI*. Dublin, Ireland : Association for Computational Linguistics, p. 21-33. DOI : 10.18653/v1/2022.nlp4convai-1.3. URL : <https://aclanthology.org/2022.nlp4convai-1.3>.
- LIN, Xiang et al. (2019). "A Unified Linear-Time Framework for Sentence-Level Discourse Parsing". In : *Annual Meeting of the Association for Computational Linguistics*.
- LIN, Zhaojiang, Andrea MADOTTO, Genta Indra WINATA et al. (2020). "MinTL : Minimalist Transfer Learning for Task-Oriented Dialogue Systems". In : *ArXiv abs/2009.12005*. URL : <https://api.semanticscholar.org/CorpusID:221949448>.
- LIN, Zhaojiang, Andrea MADOTTO, Chien-Sheng WU et al. (2019). "Personalizing Dialogue Agents via Meta-Learning". In : *Annual Meeting of the Association for Computational Linguistics*.
- LIU, Bing et Ian LANE (2016). "Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling". In : *ArXiv abs/1609.01454*. URL : <https://api.semanticscholar.org/CorpusID:7476732>.
- (2017). "Iterative policy learning in end-to-end trainable task-oriented neural dialog models". In : *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, p. 482-489. URL : <https://api.semanticscholar.org/CorpusID:1016820>.
- LIU, Cao et al. (2019). "Vocabulary Pyramid Network : Multi-Pass Encoding and Decoding with Multi-Level Vocabularies for Response Generation". In : *Annual Meeting of the Association for Computational Linguistics*.
- LIU, Qian et al. (2020). "You Impress Me : Dialogue Generation via Mutual Persona Perception". In : *Annual Meeting of the Association for Computational Linguistics*.

- LIU, Zeming et al. (2020). “Towards conversational recommendation over multi-type dialogs”. In : *arXiv preprint arXiv :2005.03954*.
- LOEPP, Benedikt, Tim HUSSEIN et Jürgen ZIEGLER (2014). “Choice-based preference elicitation for collaborative filtering recommender systems”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 3085-3094.
- LOWE, Ryan et al. (2015). “The Ubuntu Dialogue Corpus : A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems”. In : *SIGDIAL Conference*.
- LU, Jiasen, Dhruv BATRA et al. (2019). “ViLBERT : Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks”. In : *Neural Information Processing Systems*.
- LU, Jiasen, Vedanuj GOSWAMI et al. (2019). “12-in-1 : Multi-Task Vision and Language Representation Learning”. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 10434-10443.
- LUO, Kai, Scott SANNER et al. (2020). “Latent linear critiquing for conversational recommender systems”. In : *Proceedings of The Web Conference 2020*, p. 2535-2541.
- LUO, Kai, Hojin YANG et al. (2020). “Deep critiquing for VAE-based recommender systems”. In : *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, p. 1269-1278.
- MA, Wentao et al. (2020). “Conversational Word Embedding for Retrieval-Based Dialog System”. In : *ArXiv abs/2004.13249*.
- MAJUMDER, Bodhisattwa Prasad, Harsh JHAMTANI et al. (2020). “Like Hiking ? You Probably Enjoy Nature : Persona-grounded Dialog with Commonsense Expansions”. In : *Conference on Empirical Methods in Natural Language Processing*.
- MAJUMDER, Bodhisattwa Prasad, Shuyang LI et al. (2020). “Interview : Large-scale Modeling of Media Dialog with Discourse Patterns and Knowledge Grounding”. In : *Conference on Empirical Methods in Natural Language Processing*.
- MANGILI, Francesca et al. (2020). “A Bayesian approach to conversational recommendation systems”. In : *arXiv preprint arXiv :2002.05063*.
- MARCELINO, José et al. (2018). “A hierarchical deep learning natural language parser for fashion”. In : *arXiv preprint arXiv :1806.09511*.



- MAZARÉ, Pierre-Emmanuel et al. (2018). “Training Millions of Personalized Dialogue Agents”. In : *Conference on Empirical Methods in Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:52167799>.
- MEHRI, Shikib et al. (2019). “Pretraining Methods for Dialog Context Representation Learning”. In : *Annual Meeting of the Association for Computational Linguistics*.
- MESNIL, Grégoire, Yann DAUPHIN et al. (2015). “Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding”. In : *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23, p. 530-539. URL : <https://api.semanticscholar.org/CorpusID:1317136>.
- MESNIL, Grégoire, Xiaodong HE et al. (2013). “Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding”. In : *Interspeech*. URL : <https://api.semanticscholar.org/CorpusID:1701504>.
- META AI (2024). *Introducing Meta Llama 3 : The most capable openly available LLM to date*. URL : <https://ai.meta.com/blog/meta-llama-3>.
- MILLER, Alexander H. et al. (2016). “Key-Value Memory Networks for Directly Reading Documents”. In : *ArXiv abs/1606.03126*. URL : <https://api.semanticscholar.org/CorpusID:2711679>.
- MISHRA, Swaroop et al. (2021). “Cross-Task Generalization via Natural Language Crowdsourcing Instructions”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:237421373>.
- MOON, Seungwhan et al. (2019a). “OpenDialKG : Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs”. In : *Annual Meeting of the Association for Computational Linguistics*.
- (2019b). “Opendialkg : Explainable conversational reasoning with attention-based walks over knowledge graphs”. In : *Proceedings of the 57th annual meeting of the association for computational linguistics*, p. 845-854.
- MOSTAFAZADEH, N. et al. (2017). “Image-Grounded Conversations : Multimodal Context for Natural Question and Response Generation”. In : *International Joint Conference on Natural Language Processing*.
- MRKSIC, Nikola, Diarmuid Ó SÉAGHDHA, Blaise THOMSON et al. (2015). “Multi-domain Dialog State Tracking using Recurrent Neural Networks”. In : *Annual*

- Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:7597872>.
- MRKSIC, Nikola, Diarmuid Ó SÉAGHDHA, Tsung-Hsien WEN et al. (2016). “Neural Belief Tracker : Data-Driven Dialogue State Tracking”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:437687>.
- MUNIGALA, Vitobha, Srikanth TAMILSELVAM et Anush SANKARAN (2017). “”Let me convince you to buy my product...” : A Case Study of an Automated Persuasive System for Fashion Products”. In : *arXiv preprint arXiv :1709.08366*.
- NARAYANAN, Arvind, Elaine SHI et Benjamin I. P. RUBINSTEIN (2011). “Link prediction by de-anonymization : How We Won the Kaggle Social Network Challenge”. In : *The 2011 International Joint Conference on Neural Networks*, p. 1825-1834. DOI : 10.1109/IJCNN.2011.6033446.
- NI, Jinjie et al. (2021). “Recent advances in deep learning based dialogue systems : a systematic survey”. In : *Artificial Intelligence Review* 56, p. 3055-3155.
- NOVIKOVA, Jekaterina, Ondrej DUSEK et Verena RIESER (2017). “The E2E Dataset : New Challenges For End-to-End Generation”. In : *ArXiv abs/1706.09254*.
- OPENAI (2023). “GPT-4 Technical Report”. In : *PREPRINT*.
- OPENAI et al. (2024). *GPT-4 Technical Report*. arXiv : 2303.08774 [cs.CL].
- ORABY, Shereen et al. (2019). “Curate and Generate : A Corpus and Method for Joint Control of Semantics and Style in Neural NLG”. In : *ArXiv abs/1906.01334*.
- OUYANG, Long et al. (2022). “Training language models to follow instructions with human feedback”. In : *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. New Orleans, LA, USA : Curran Associates Inc. ISBN : 9781713871088.
- PAL, Arka et al. (2024). “Smaug : Fixing Failure Modes of Preference Optimisation with DPO-Positive”. In : *arXiv preprint arXiv :2402.13228*.
- PENG, Baolin et al. (2023). *Check Your Facts and Try Again : Improving Large Language Models with External Knowledge and Automated Feedback*. arXiv : 2302.12813 [cs.CL].
- PENG, Bo et al. (2024). “eCeLLM : Generalizing Large Language Models for E-commerce from Large-scale, High-quality Instruction Data”. In : *arXiv preprint arXiv :2402.08831*.

- PU, Pearl et Boi FALTINGS (2004). “Decision tradeoff using example-critiquing and constraint programming”. In : *Constraints* 9, p. 289-310.
- QI, Di et al. (2020). “ImageBERT : Cross-modal Pre-training with Large-scale Weak-supervised Image-Text Data”. In : *ArXiv* abs/2001.07966.
- QIAN, Kun et Zhou YU (2019). “Domain Adaptive Dialog Generation via Meta Learning”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:182952555>.
- QUAN, Jun et Deyi XIONG (2020). “Modeling Long Context for Task-Oriented Dialogue State Generation”. In : *ArXiv* abs/2004.14080. URL : <https://api.semanticscholar.org/CorpusID:216641784>.
- QUAN, Jun, Shian ZHANG et al. (2020). “RiSAWOZ : A Large-Scale Multi-Domain Wizard-of-Oz Dataset with Rich Semantic Annotations for Task-Oriented Dialogue Modeling”. In : *Conference on Empirical Methods in Natural Language Processing*.
- QWEN TEAM (2024). *Introducing Qwen1.5*. URL : <https://qwenlm.github.io/blog/qwen1.5>.
- RADFORD, Alec et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. arXiv : 2103.00020 [cs.CV]. URL : <https://arxiv.org/abs/2103.00020>.
- RAFFEL, Colin et al. (2019). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In : *J. Mach. Learn. Res.* 21, 140 :1-140 :67. URL : <https://api.semanticscholar.org/CorpusID:204838007>.
- RAJPURKAR, Pranav, Robin JIA et Percy LIANG (2018). “Know What You Don’t Know : Unanswerable Questions for SQuAD”. In : *ArXiv* abs/1806.03822. URL : <https://api.semanticscholar.org/CorpusID:47018994>.
- RASTOGI, Abhinav et al. (2019). “Towards Scalable Multi-domain Conversational Agents : The Schema-Guided Dialogue Dataset”. In : *AAAI Conference on Artificial Intelligence*.
- (2020). “Towards scalable multi-domain conversational agents : The schema-guided dialogue dataset”. In : 34.05, p. 8689-8696.
- RAUX, Antoine et al. (2005). “Let’s go public ! taking a spoken dialog system to the real world”. In : *Interspeech*. URL : <https://api.semanticscholar.org/CorpusID:281507>.

- RAVURI, Suman V. et Andreas STOLCKE (2015). "Recurrent neural network and LSTM models for lexical utterance classification". In : *Interspeech*. URL : <https://api.semanticscholar.org/CorpusID:7639893>.
- (2016). "A comparative study of recurrent neural network models for lexical domain classification". In : *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 6075-6079. URL : <https://api.semanticscholar.org/CorpusID:10687996>.
- REDDY, Chandan K et al. (2022). "Shopping queries dataset : A large-scale ESCI benchmark for improving product search". In : *arXiv preprint arXiv :2206.06588*.
- REITER, Ehud (1994). "Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible?" In : *ArXiv abs/cmp-lg/9411032*. URL : <https://api.semanticscholar.org/CorpusID:2652169>.
- REN, Hang et al. (2013). "Dialog State Tracking using Conditional Random Fields". In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:11344817>.
- ROJAS-BARAHONA, Lina Maria et al. (2016). "A Network-based End-to-End Trainable Task-oriented Dialogue System". In : *Conference of the European Chapter of the Association for Computational Linguistics*.
- ROLLER, Stephen et al. (2020). "Recipes for Building an Open-Domain Chatbot". In : *Conference of the European Chapter of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:216562425>.
- ROSTAMZADEH, Negar et al. (2018). "Fashion-Gen : The Generative Fashion Dataset and Challenge". In : *ArXiv abs/1806.08317*. URL : <https://api.semanticscholar.org/CorpusID:49343047>.
- SAHA, Amrita, Mitesh M. KHAPRA et Karthik SANKARANARAYANAN (2017). "Towards Building Large Scale Multimodal Domain-Aware Conversation Systems". In : *AAAI Conference on Artificial Intelligence*.
- (2018). "Towards Building Large Scale Multimodal Domain-Aware Conversation Systems". In.
- SAHA, Sougata, Souvik DAS et Rohini SRIHARI (mai 2022). "Stylistic Response Generation by Controlling Personality Traits and Intent". In : *Proceedings of the 4th Workshop on NLP for Conversational AI*. Dublin, Ireland : Association for

- Computational Linguistics, p. 197-211. DOI : 10.18653/v1/2022.nlp4convai-1.16. URL : <https://aclanthology.org/2022.nlp4convai-1.16>.
- SANH, Victor et al. (2019). “DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter”. In : *ArXiv abs/1910.01108*. URL : <https://api.semanticscholar.org/CorpusID:203626972>.
- SANKAR, Chinnadhurai et al. (2019). “Do Neural Dialog Systems Use the Conversation History Effectively ? An Empirical Study”. In : *Annual Meeting of the Association for Computational Linguistics*.
- SCHNABEL, Tobias et al. (2018). “Short-term satisfaction and long-term coverage : Understanding how users tolerate algorithmic exploration”. In : *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, p. 513-521.
- SEKULIĆ, Ivan, Mohammad ALIANNEJADI et Fabio CRESTANI (2022). “Evaluating Mixed-Initiative Conversational Search Systems via User Simulation”. In : *WSDM '22. Virtual Event, AZ, USA : Association for Computing Machinery*, p. 888-896. ISBN : 9781450391320. DOI : 10.1145/3488560.3498440. URL : <https://doi.org/10.1145/3488560.3498440>.
- SEO, Paul Hongsuck et al. (2017). “Visual Reference Resolution using Attention Memory for Visual Dialog”. In : *NIPS*.
- SEVEGNANI, Karin et al. (2021). “OTTers : One-turn Topic Transitions for Open-Domain Dialogue”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:235248336>.
- SHAH, Pararth et al. (2018). “Building a Conversational Agent Overnight with Dialogue Self-Play”. In : *ArXiv abs/1801.04871*.
- SHAN, Yong et al. (2020). “A Contextual Hierarchical Attention Network with Adaptive Objective for Dialogue State Tracking”. In : *ArXiv abs/2006.01554*. URL : <https://api.semanticscholar.org/CorpusID:219179747>.
- SHUSTER, Kurt, Samuel HUMEAU et al. (2018). “Image-Chat : Engaging Grounded Conversations”. In : *Annual Meeting of the Association for Computational Linguistics*.
- SHUSTER, Kurt, Da JU et al. (2019). “The Dialogue Dodecathlon : Open-Domain Knowledge and Image Grounded Conversational Agents”. In : *Annual Meeting of the Association for Computational Linguistics*.

- SINGH, Amanpreet, Vedanuj GOSWAMI et Devi PARIKH (2020). “Are we pre-training it right ? Digging deeper into visio-linguistic pretraining”. In : *ArXiv abs/2004.08744*.
- SMITH, Eric Michael et al. (2020). “Can You Put it All Together : Evaluating Conversational Agents’ Ability to Blend Skills”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:215827653>.
- SMITH, Shaden et al. (2022). “Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model”. In : *ArXiv abs/2201.11990*. URL : <https://api.semanticscholar.org/CorpusID:246411325>.
- SMYTH, Barry et Lorraine MCGINTY (2003). “An analysis of feedback strategies in conversational recommenders”. In : *the Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS 2003)*. Citeseer.
- SONG, Haoyu, Yan WANG, Weinan ZHANG, Xiaojiang LIU et al. (2020). “Generate, Delete and Rewrite : A Three-Stage Framework for Improving Persona Consistency of Dialogue Generation”. In : *ArXiv abs/2004.07672*.
- SONG, Haoyu, Yan WANG, Weinan ZHANG, Zhengyu ZHAO et al. (2020). “Profile Consistency Identification for Open-domain Dialogue Agents”. In : *Conference on Empirical Methods in Natural Language Processing*.
- SONG, Zhenqiao et al. (2019). “Generating Responses with a Specific Emotion in Dialog”. In : *Annual Meeting of the Association for Computational Linguistics*.
- SU, Pei-hao et al. (2016). “Continuously Learning Neural Dialogue Management”. In : *ArXiv abs/1606.02689*. URL : <https://api.semanticscholar.org/CorpusID:16752980>.
- SU, Shang-Yu, Chao-Wei HUANG et Yun-Nung CHEN (2019). “Dual Supervised Learning for Natural Language Understanding and Generation”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:155092173>.
- SU, Weijie et al. (2019). “VL-BERT : Pre-training of Generic Visual-Linguistic Representations”. In : *ArXiv abs/1908.08530*.

- SUN, Kai et al. (2020). "Adding Chit-Chat to Enhance Task-Oriented Dialogues". In : *North American Chapter of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:225068315>.
- SUN, Weiwei et al. (2021). "Simulating user satisfaction for the evaluation of task-oriented dialogue systems". In : *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 2499-2506.
- SUN, Yueming et Yi ZHANG (2018). "Conversational recommender system". In : *The 41st international acm sigir conference on research & development in information retrieval*, p. 235-244.
- SUNDARARAJ, Vinu et M R REJEESH (2021). "A detailed behavioral analysis on consumer and customer changing behavior with respect to social networking sites". In : *Journal of Retailing and Consumer Services* 58, p. 102190. ISSN : 0969-6989. DOI : <https://doi.org/10.1016/j.jretconser.2020.102190>. URL : <https://www.sciencedirect.com/science/article/pii/S0969698920306238>.
- SUTSKEVER, Ilya, Oriol VINYALS et Quoc V. LE (2014). "Sequence to Sequence Learning with Neural Networks". In : *ArXiv abs/1409.3215*.
- TAKANOBU, Ryuichi, Runze LIANG et Minlie HUANG (2020). "Multi-Agent Task-Oriented Dialog Policy Learning with Role-Aware Reward Decomposition". In : *ArXiv abs/2004.03809*. URL : <https://api.semanticscholar.org/CorpusID:215415923>.
- TAKMAZ, Ece et al. (2020). "Refer, Reuse, Reduce : Generating Subsequent References in Visual and Conversational Contexts". In : *ArXiv abs/2011.04554*.
- TAN, Hao Hao et Mohit BANSAL (2019). "LXMERT : Learning Cross-Modality Encoder Representations from Transformers". In : *Conference on Empirical Methods in Natural Language Processing*.
- TANG, Jianheng et al. (2019). "Target-Guided Open-Domain Conversation". In : *ArXiv abs/1905.11553*.
- TAORI, Rohan et al. (2023). *Stanford Alpaca : An Instruction-following LLaMA model*. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).
- THOMSON, Blaise, Jost SCHATZMANN et Steve J. YOUNG (2008). "Bayesian update of dialogue state for robust dialogue systems". In : *2008 IEEE International*

- Conference on Acoustics, Speech and Signal Processing*, p. 4937-4940. URL : <https://api.semanticscholar.org/CorpusID:10412895>.
- TIAN, Junfeng et al. (2023). *ChatPLUG : Open-Domain Generative Dialogue System with Internet-Augmented Instruction Tuning for Digital Human*. arXiv : 2304.07849 [cs.CL].
- TOU, Frederich N et al. (1982). "RABBIT : An Intelligent Database Assistant." In : *AAAI*. T. 82. Citeseer, p. 314-318.
- TOUVRON, Hugo, Thibaut LAVRIL et al. (2023). "LLaMA : Open and Efficient Foundation Language Models". In : *ArXiv abs/2302.13971*. URL : <https://api.semanticscholar.org/CorpusID:257219404>.
- TOUVRON, Hugo, Louis MARTIN et al. (2023). "Llama 2 : Open Foundation and Fine-Tuned Chat Models". In : *ArXiv abs/2307.09288*. URL : <https://api.semanticscholar.org/CorpusID:259950998>.
- TRAN, Van-Khanh et Le-Minh NGUYEN (2017). "Semantic Refinement GRU-Based Neural Language Generation for Spoken Dialogue Systems". In : *International Conference of the Pacific Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:3820239>.
- TÜR, Gökhan, Dilek Z. HAKKANI-TÜR et Larry HECK (2010). "What is left to be understood in ATIS?" In : *2010 IEEE Spoken Language Technology Workshop*, p. 19-24.
- TVERSKY, Amos et Daniel KAHNEMAN (1992). "Advances in prospect theory : Cumulative representation of uncertainty". In : *Journal of Risk and Uncertainty* 5, p. 297-323.
- TVERSKY, Amos et Itamar SIMONSON (1993). "Context-dependent preferences". In : *Management science* 39.10, p. 1179-1189.
- VASILEVA, Mariya I. et al. (2018). "Learning Type-Aware Embeddings for Fashion Compatibility". In : *ArXiv abs/1803.09196*. URL : <https://api.semanticscholar.org/CorpusID:4405489>.
- VASWANI, Ashish et al. (2017). "Attention is All you Need". In : *NIPS*.
- VENDROV, Ivan et al. (2020). "Gradient-based optimization for bayesian preference elicitation". In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 34. 06, p. 10292-10301.



- VRIES, Harm de et al. (2016). “GuessWhat ?! Visual Object Discovery through Multi-modal Dialogue”. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 4466-4475.
- WANG, Huazheng, Qingyun WU et Hongning WANG (2017). “Factorization bandits for interactive recommendation”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 31. 1.
- WANG, Kai et al. (2020). “Multi-Domain Dialogue Acts and Response Co-Generation”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:216553145>.
- WANG, Weikang et al. (2019). “Incremental Learning from Scratch for Task-Oriented Dialogue Systems”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:186206602>.
- WANG, Xuewei et al. (2019). “Persuasion for good : Towards a personalized persuasive dialogue system for social good”. In : *arXiv preprint arXiv :1906.06725*.
- WANG, Ye, Yi GUO et Siqi ZHU (2020). “Slot Attention with Value Normalization for Multi-domain Dialogue State Tracking”. In : *Conference on Empirical Methods in Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:226262220>.
- WANG, Zhiguo, Wael HAMZA et Radu FLORIAN (2017). “Bilateral Multi-Perspective Matching for Natural Language Sentences”. In : *International Joint Conference on Artificial Intelligence*. URL : <https://api.semanticscholar.org/CorpusID:9395040>.
- WEI, Jason et al. (2022). “Chain-of-thought prompting elicits reasoning in large language models”. In : *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. New Orleans, LA, USA : Curran Associates Inc. ISBN : 9781713871088.
- WEN, Tsung-Hsien, Milica GASIC, Dongho KIM et al. (2015). “Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking”. In : *ArXiv abs/1508.01755*. URL : <https://api.semanticscholar.org/CorpusID:1139492>.
- WEN, Tsung-Hsien, Milica GASIC, Nikola MRKSIC, Lina Maria ROJAS-BARAHONA, Pei-hao SU, Stefan ULTES et al. (2016). “Conditional Generation and Snapshot

- Learning in Neural Dialogue Systems”. In : *Conference on Empirical Methods in Natural Language Processing*.
- WEN, Tsung-Hsien, Milica GASIC, Nikola MRKSIC, Lina Maria ROJAS-BARAHONA, Pei-hao SU, David VANDYKE et al. (2016). “Multi-domain Neural Network Language Generation for Spoken Dialogue Systems”. In : *ArXiv abs/1603.01232*. URL : <https://api.semanticscholar.org/CorpusID:6508854>.
- WHITE, Jules et al. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. arXiv : 2302.11382 [cs.SE].
- WILLIAMS, J. (2014). “Web-style ranking and SLU combination for dialog state tracking”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:765934>.
- WILLIAMS, J., Kavosh ASADI et Geoffrey ZWEIG (2017). “Hybrid Code Networks : practical and efficient end-to-end dialog control with supervised and reinforcement learning”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:13214003>.
- WILLIAMS, J., Antoine RAUX et al. (2013). “The Dialog State Tracking Challenge”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:10250499>.
- WILLIAMS, J. et Geoffrey ZWEIG (2016). “End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning”. In : *ArXiv abs/1606.01269*. URL : <https://api.semanticscholar.org/CorpusID:12923236>.
- WILLIAMS, Jason (2013). “Multi-domain learning and generalization in dialog state tracking”. In : *SIGDIAL Conference*. URL : <https://api.semanticscholar.org/CorpusID:1175358>.
- WU, Chien-Sheng, Steven C. H. HOI et al. (2020). “TOD-BERT : Pre-trained Natural Language Understanding for Task-Oriented Dialogue”. In : *Conference on Empirical Methods in Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:215768835>.
- WU, Chien-Sheng, Andrea MADOTTO et al. (2019). “Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems”. In : *Annual Meeting of the Association for Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:160009896>.

- WU, Felix et al. (2019). "Simplifying Graph Convolutional Networks". In : *International Conference on Machine Learning*. URL : <https://api.semanticscholar.org/CorpusID:67752026>.
- WU, Ga et al. (2019). "Deep language-based critiquing for recommender systems". In : *Proceedings of the 13th ACM Conference on Recommender Systems*, p. 137-145.
- WU, Hui et al. (2021a). "Fashion iq : A new dataset towards retrieving images by natural language feedback". In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 11307-11317.
- (2021b). "The Fashion IQ Dataset : Retrieving Images by Combining Side Information and Relative Natural Language Feedback". In : *CVPR*.
- WU, Jiawei, Xin Eric WANG et William Yang WANG (2019). "Self-Supervised Dialogue Learning". In : *ArXiv abs/1907.00448*.
- WU, Wenquan et al. (2019). "Proactive human-machine conversation with explicit conversation goals". In : *arXiv preprint arXiv :1906.05572*.
- XIAO, Han, Georgios MASTRAPAS et Bo WANG (2024). "Jina CLIP : Your CLIP Model Is Also Your Text Retriever". In : *Multi-modal Foundation Model meets Embodied AI Workshop @ ICML2024*. URL : <https://openreview.net/forum?id=1SDkG98goM>.
- XU, Hu et al. (2020). "User memory reasoning for conversational recommendation". In : *arXiv preprint arXiv :2006.00184*.
- XU, Kerui et al. (2021). "Adapting user preference to online feedback in multi-round conversational recommendation". In : *Proceedings of the 14th ACM international conference on web search and data mining*, p. 364-372.
- YANG, An et al. (2024). "Qwen2 technical report". In : *arXiv preprint arXiv :2407.10671*.
- YANG, Hojin et al. (2021). "Bayesian Preference Elicitation with Keyphrase-Item Coembeddings for Interactive Recommendation". In : *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, p. 55-64.
- YANG, Jingfeng et al. (2023). "Harnessing the Power of LLMs in Practice : A Survey on ChatGPT and Beyond". In : *ACM Transactions on Knowledge Discovery from Data* 18, p. 1-32. URL : <https://api.semanticscholar.org/CorpusID:258331833>.

- YANG, Shiquan, Rui ZHANG et Sarah Monazam ERFANI (2020). “GraphDialog : Integrating Graph Knowledge into End-to-End Task-Oriented Dialogue Systems”. In : *ArXiv abs/2010.01447*. URL : <https://api.semanticscholar.org/CorpusID:222133138>.
- YANG, Xuewen et al. (2020). “Fashion Captioning : Towards Generating Accurate Descriptions with Semantic Rewards”. In : *European Conference on Computer Vision*. URL : <https://api.semanticscholar.org/CorpusID:221006339>.
- YAO, Kaisheng, Baolin PENG et al. (2014). “Spoken language understanding using long short-term memory neural networks”. In : *2014 IEEE Spoken Language Technology Workshop (SLT)*, p. 189-194. URL : <https://api.semanticscholar.org/CorpusID:1281039>.
- YAO, Kaisheng, Geoffrey ZWEIG et al. (2013). “Recurrent neural networks for language understanding”. In : *Interspeech*. URL : <https://api.semanticscholar.org/CorpusID:9152760>.
- YING, Rex et al. (2018). “Graph convolutional neural networks for web-scale recommender systems”. In : *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, p. 974-983.
- YOUNG, Steve J. et al. (2010). “The Hidden Information State model : A practical framework for POMDP-based spoken dialogue management”. In : *Comput. Speech Lang.* 24, p. 150-174. URL : <https://api.semanticscholar.org/CorpusID:13302682>.
- YU, Fei et al. (2020). “ERNIE-ViL : Knowledge Enhanced Vision-Language Representations Through Scene Graph”. In : *ArXiv abs/2006.16934*.
- YU, Tao et Shafiq R. JOTY (2020). “Online Conversation Disentanglement with Pointer Networks”. In : *Conference on Empirical Methods in Natural Language Processing*.
- YU, Tong, Yilin SHEN et Hongxia JIN (2019). “A visual dialog augmented interactive recommender system”. In : *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, p. 157-165.
- YUAN, Yifei et Wai LAM (2021). “Conversational Fashion Image Retrieval via Multiturn Natural Language Feedback”. In : *Proceedings of the 44th In-*

- ternational ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 839-848.
- ZENG, Guangtao et al. (nov. 2020). “MedDialog : Large-scale Medical Dialogue Datasets”. In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, p. 9241-9250. DOI : 10.18653/v1/2020.emnlp-main.743. URL : <https://aclanthology.org/2020.emnlp-main.743>.
- ZHANG, Saizheng et al. (2018). “Personalizing Dialogue Agents : I have a dog, do you have pets too ?” In : *ArXiv abs/1801.07243*.
- ZHANG, Shuo et Krisztian BALOG (2020). “Evaluating conversational recommender systems via user simulation”. In : *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, p. 1512-1520.
- ZHANG, Xiaoying et al. (2020). “Conversational contextual bandit : Algorithm and application”. In : *Proceedings of the web conference 2020*, p. 662-672.
- ZHANG, Yichi et al. (2020). “A Probabilistic End-To-End Task-Oriented Dialog Model with Latent Belief States towards Semi-Supervised Learning”. In : *Conference on Empirical Methods in Natural Language Processing*. URL : <https://api.semanticscholar.org/CorpusID:221761320>.
- ZHANG, Yongfeng et al. (2018). “Towards conversational search and recommendation : System ask, user respond”. In : *Proceedings of the 27th acm international conference on information and knowledge management*, p. 177-186.
- ZHANG, Zhuosheng et al. (2018). “Modeling Multi-turn Conversation with Deep Utterance Aggregation”. In : *International Conference on Computational Linguistics*.
- ZHAO, Tiancheng et Maxine ESKÉNAZI (2016). “Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning”. In : *ArXiv abs/1606.02560*. URL : <https://api.semanticscholar.org/CorpusID:6179947>.
- (2018). “Zero-Shot Dialog Generation with Cross-Domain Latent Actions”. In : *SIGDIAL Conference*.
- ZHAO, Wayne Xin et al. (2023). *A Survey of Large Language Models*. arXiv : 2303.18223 [cs.CL].

- ZHAO, Xiaoxue, Weinan ZHANG et Jun WANG (2013). "Interactive collaborative filtering". In : *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, p. 1411-1420.
- ZHENG, Yinhe et al. (2019). "Persona-aware dialogue generation with enriched profile". In : *Proc. 33rd Conf. Neural Inf. Process. Syst.(NIPS)*, p. 1-10.
- ZHOU, Guorui et al. (2017). "Deep Interest Network for Click-Through Rate Prediction". In : *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. URL : <https://api.semanticscholar.org/CorpusID:1637394>.
- ZHOU, Hao, Minlie HUANG et Xiaoyan ZHU (2016). "Context-aware Natural Language Generation for Spoken Dialogue Systems". In : *International Conference on Computational Linguistics*. URL : <https://api.semanticscholar.org/CorpusID:17002591>.
- ZHOU, Kun, Wayne Xin ZHAO et al. (2020). "Improving conversational recommender systems via knowledge graph based semantic fusion". In : *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, p. 1006-1014.
- ZHOU, Kun, Yuanhang ZHOU et al. (2020). "Towards topic-guided conversational recommender system". In : *arXiv preprint arXiv :2010.04125*.
- ZHOU, Luwei et al. (2019). "Unified Vision-Language Pre-Training for Image Captioning and VQA". In : *ArXiv abs/1909.11059*.
- ZHOU, Sijin et al. (2020). "Interactive recommender system via knowledge graph-enhanced reinforcement learning". In : *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, p. 179-188.
- ZHU, Deyao et al. (2023). "MiniGPT-4 : Enhancing Vision-Language Understanding with Advanced Large Language Models". In : *ArXiv abs/2304.10592*. URL : <https://api.semanticscholar.org/CorpusID:258291930>.
- ZHU, Han et al. (2018). "Learning Tree-based Deep Model for Recommender Systems". In : *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. URL : <https://api.semanticscholar.org/CorpusID:25648541>.

- ZHU, Qingfu et al. (2018). “Retrieval-Enhanced Adversarial Training for Neural Response Generation”. In : *Annual Meeting of the Association for Computational Linguistics*.
- ZILKA, Lukás et Filip JURČÍČEK (2015). “Incremental LSTM-based dialog state tracker”. In : *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, p. 757-762. URL : <https://api.semanticscholar.org/CorpusID:18436668>.
- ZOU, Lixin et al. (2020). “Neural interactive collaborative filtering”. In : *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 749-758.
- ZOU, Xingxing et al. (2019). “Fashionai : A hierarchical dataset for fashion understanding”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, p. 0-0.
- ZUCCON, Guido et Bevan KOOPMAN (2023). *Dr ChatGPT, tell me what I want to hear : How prompt knowledge impacts health answer correctness*. arXiv : 2302.13793 [cs.CL].





# Annexe A

## Annexe

### A.1 Résumé des ensembles de données

TAB. A.1 : Jeux de données *TOD*

Titre	Domaines	Objectif	Méthode	Citation
Clinical Conversations	Symptômes et états dans conversations cliniques	<i>NLU</i>	Collection des véritables conversations et annotation humaine.	N. DU et al. 2019
CLINC150	150+ intentions connues et inconnues	<i>NLU</i>	Collection des requêtes en crowdsourcing.	LARSON et al. 2019
ATIS	Réservation des billets d'avion	<i>NLU, DST</i>	Collection des véritables appels et annotation humaine.	TÜR, HAKKANI-TÜR et HECK 2010
E2E	Description des restaurants	<i>NLG</i>	Collection des descriptions en AMT crowdsourcing.	NOVIKOVA, DUSEK et RIESER 2017
Weather Forecast	Discours de prévision météorologique	<i>NLG</i>		BALAKRISHNAN et al. 2019

YelpNLG	Commentaires sur des restaurants	NLG	Collecte des véritables commentaires et extraction automatique.	ORABY et al. 2019
*bAbI	Restaurants	TOD	Simulation plus collection des véritables dialogues entre humain et machine.	BORDES et WESTON 2016
Ubuntu	Informatique (Ubuntu)	TOD	Collecte des véritables questions-réponses sur le forum	LOWE et al. 2015
Banking	E-banking	TOD	Collecte des véritables questions-réponses sur les pages web	M. HENDERSON, VULIC et al. 2019
*SimDial	Restaurant, cinéma, bus et météo	TOD	Simulation	T. ZHAO et ESKÉNAZI 2018
WOZ	Restaurant	TOD	Collecte des dialogues Humain-Humain par crowdsourcing en AMT	ROJAS-BARAHONA et al. 2016
CamRest	Restaurant	TOD	Collecte des dialogues par la méthode WoZ en AMT	T.-H. WEN, GASIC, MRKSIC, ROJAS-BARAHONA, P.-h. SU, ULTES et al. 2016

SMD	Météo, navigation, planification	<i>TOD</i>	Collecte des dialogues par la méthode WoZ en AMT	ERIC et al. 2017
MultiWOZ	Restaurant, bus, alarme, etc	<i>TOD</i>	Collecte des dialogues par la méthode WoZ en AMT	BUDZIANOWSKI et al. 2018
Frames	Voyage	<i>TOD</i>	Collecte des dialogues par la méthode WoZ	ASRI et al. 2017
MSR-E2E	Réservation de billets de cinéma, restaurant et taxi.	<i>TOD</i>	Collecte des dialogues par la méthode WoZ.	X. LI, PANDA et al. 2018
MetaLWOZ	47 domaines et 227 tâches	<i>TOD</i>	Collecte des dialogues par la méthode WoZ en AMT	S. LEE, SCHULZ et al. 2019
Taskmaster	6 domaines	<i>TOD</i>	Collecte des dialogues par la méthode WoZ en crowdsourcing	BYRNE et al. 2019
MedDialog (EN,CN)	Médical	<i>TOD</i>	Collecte des dialogues réels en ligne	ZENG et al. 2020
Court Debate Dataset	Judicial	<i>TOD</i>	Collecte des dialogues réels	JI et al. 2020
JDDC	E-commerce	<i>TOD</i>	Collecte des dialogues réels	M. CHEN et al. 2019
TreeDST	10 domaines	<i>TOD</i>	Simulation puis reformulation humaine	J. CHENG et al. 2020
RiSAWOZ	12 domaines en chinois	<i>TOD</i>	Collection des dialogues par la méthode WoZ	QUAN, Shian ZHANG et al. 2020

SB-TOP	Alarme, météo, appel et musique	TOD	Collection des sessions écrites en crowdsourcing et des annotations sémantiques	AGHAJANYAN et al. 2020
*GSIM	Réservation des restaurants et des cinémas	TOD	Simulation puis reformulation humaine	SHAH et al. 2018
cite-8K	Réservation des restaurants	TOD	Collecte des appels réels	COOPE et al. 2020
SGD	16 domaines	TOD	Simulation puis reformulation humaine	RASTOGI et al. 2019
E-commerce Dialogue Corpus	Service client	TOD	Collecte des dialogues réels	Z. ZHANG et al. 2018
PersonaChat	Bavardage	OOD, Personalisation	Collection des persona et des dialogues WOZ	Saizheng ZHANG et al. 2018
PersonaChat Gen	Bavardage	OOD, Personalisation	Simulation des personas et des conversations par GPT3	Y.-J. LEE, LIM et al. 2022
SalesBot	Bavardage, cinéma	OOD, TOD	Simulation puis reformulation par les modèles de langage larges	CHIU et al. 2022

KvPI	Bavardage	OOD, Per- son- nali- sation	Collection des perso- na et des réponses sur internet et anno- tation humaine	H. SONG, Yan WANG, W. ZHANG, Z. ZHAO et al. 2020
ConvAI	Bavardage	OOD, Per- son- nali- sation	Complément en moyen de WOZ à la base de PersonaChat	DINAN et al. 2019
GuessWhat ?!	Recognition des objets	OOD, Multi- modal	Collection des dia- logues par crowd- sourcing	VRIES et al. 2016
VisDial	Images et photos du jeu de données COCO	OOD, Multi- modal	Collection des dia- logues par crowd- sourcing	A. DAS et al. 2016
CLEVR- Dialog	Images des ob- jets dans dif- férents arrange- ments des posi- tions	Multimodal, Posi- tion des objets	Collection des dia- logues par crowd- sourcing	KOTTUR, MOURA et al. 2019
MNIST Dia- log	Images des chiffres écrits en différentes couleurs	Multimodal	Collection des dia- logues par crowd- sourcing	SEO et al. 2017
DialogCC	Domaine général	OOD, Multi- modal	Filtrage des dia- logues source et in- sertion automatique des images par CLIP	Y.-J. LEE, Ko et al. 2022
VQA	Domaine général	Multimodal	Collection des dia- logues par crowd- sourcing	AGRAWAL et al. 2015

PhotoBook	Domaine général	<i>TOD</i> , Multi-modal	Collection des dialogues par crowdsourcing	HABER et al. 2019
IGC	Domaine général	<i>OOD</i> , Multi-modal	Collection des dialogues par crowdsourcing	MOSTAFAZADEH et al. 2017
ImageChat	Domaine général	<i>OOD</i> , Multi-modal, Personalisation	Collection des dialogues par crowdsourcing et annotation humaine	SHUSTER, HUMEAU et al. 2018
MMD	E-commerce de la mode	<i>TOD</i> , Multi-modal		A. SAHA, KHAPRA et SANKARANARAYANAN 2017
SIMMC	E-commerce de la mode et meubles	<i>TOD</i> , Multi-modal	Collection des dialogues par crowdsourcing	CROOK et al. 2019
MMConv	Hôtel, centre commercial, etc	<i>TOD</i> , Multi-modal	Collection des dialogues en crowdsourcing	L. LIAO et al. 2021
MMDialog	Domaine général	<i>OOD</i> , Multi-modal	Collection des dialogues réels	FENG et al. 2022

## A.2 Évaluation par annotateurs dans la section 6.1.2

Nous présentons un aperçu des annotations des dialogues dans Section 6.1.2 (voir Figure A.1). Les scores sont noté pour chaque tour de parole.

### A.3 Démonstration et étude de cas dans la section 6.1.3

dialogueID	N Tour	payé?	info objective	info subjective	Préf/Goal non-conflit	patience correcte	réceptive non-conflit
9	9	N	[0,0,0,0,1,1,0,0]	[0,0,0,1,0,1,1,0,0]	[1,1,1,1,1,1,1,1]	[1,1,1,1,1,1,1,1]	[1,1,1,1,1,1,1,1]
10	5	N	[0,0,0,0,0]	[1,0,0,0,0]	[1,1,1,1]	[1,1,1,1]	[1,1,1,1]
11	4	N	[1,0,0,0]	[1,0,0,0]	[1,1,1,1]	[1,1,1,1]	[1,1,1,1]
12	6	N	[0,0,0,0,0,0]	[0,0,0,0,0,0]	[1,1,1,1,1,1]	[1,0,1,1,1,1]	[1,1,1,1,1,1]
13	7	N	[0,0,0,0,0,1,0]	[1,1,0,1,1,0]	[1,1,1,1,1,1]	[1,1,1,1,1,0]	[1,1,1,1,1,1]
14	3	Y	[0,1,0]	[0,1,0]	[1,1,1]	[1,1,1]	[1,1,1]
15	4	N	[0,0,0,0]	[0,1,0,0]	[1,1,1,1]	[1,1,1,1]	[1,1,1,1]
16	11	Y	[0,0,0,0,0,0,0,0,0,0,0]	[0,1,1,1,1,0,0,0,0,0,0]	[1,1,1,1,1,1,1,1,1,1]	[0,1,1,1,0,0,1,1,1,1]	[1,1,0,0,1,1,1,1,1,1]
17	7	N	[0,1,0,0,0,0]	[0,1,1,0,0,0]	[1,1,1,1,1,1]	[1,1,1,1,0,0]	[1,1,1,1,1,1]
18	8	Y	[1,1,0,1,0,0,0]	[1,1,0,1,1,0,0,0]	[1,1,1,1,1,1,1,1]	[1,1,1,1,1,1,1,1]	[1,1,1,1,1,1,1,1]
19	5	N	[0,0,0,0,0]	[0,1,1,0,0]	[1,1,1,1,1]	[1,0,0,1,0]	[1,1,1,1,1]
20	8	N	[1,1,1,0,1,0,0,0]	[1,1,1,0,1,1,0,0]	[1,1,1,1,0,0,0,0]	[0,0,1,0,0,0,0]	[1,1,1,1,0,0,0,0]
21	6	Y	[0,0,0,0,0,0]	[0,1,0,0,0,0]	[1,1,1,1,1,1]	[0,0,1,1,1,0]	[1,1,1,1,1,1]
22	5	N	[0,1,0,0,0]	[0,1,0,1,1]	[1,1,1,1,1]	[1,1,1,1]	[1,1,1,1,0]
23	5	N	[1,1,0,0,0]	[1,1,0,0,0]	[1,1,1,1,1]	[1,0,1,0,1]	[1,1,1,1,1]
24	4	Y	[0,0,1,0]	[0,1,1,0]	[1,1,1,1]	[0,1,1,1]	[1,1,1,1]
25	5	N	[1,1,1,1,1]	[1,1,1,1,1]	[1,1,1,1,1]	[1,1,1,1,1]	[1,0,0,0,0]
26	9	Y	[0,0,1,0,1,0,0,0,1]	[0,1,1,1,0,0,0,1]	[1,1,1,1,1,1,1,1]	[0,0,1,0,0,1,1,1]	[1,1,1,1,1,1,1,1]
27	6	Y	[0,1,0,0,0,0]	[0,1,1,1,1,0]	[1,1,1,1,1,1]	[0,1,1,0,1,1]	[1,1,1,1,1,1]

FIG. A.1 : Les notes d'un annotateur sur les dialogues générés dans Section 6.1.2.

## A.3 Démonstration et étude de cas dans la section 6.1.3

### Cas d'usage du modèle Mistral-7b, zéro-shot

**Client** : I want find a dress for a fine dinner dating, can you recommend for me ?

**Assistant** : That sounds like a wonderful occasion ! Here are a few tips to help you find the perfect dress for a fine dinner date :

Consider the Venue : Think about the restaurant or location where you'll be having dinner. Is it formal or semi-formal ? A high-end restaurant may call for a more elegant dress, while a more relaxed venue might allow for a chic but slightly less formal outfit.

Choose the Right Style :

Little Black Dress (LBD) : A classic and versatile choice that can be dressed up or down with accessories.

Cocktail Dress : Perfect for a fine dinner, a cocktail dress usually falls just above the knee and is both elegant and stylish.

Maxi Dress : For a more romantic and flowing look, a maxi dress can be a great option.

Sheath Dress : Sleek and sophisticated, a sheath dress is perfect for a more formal setting.

Fabric and Fit : Opt for high-quality fabrics like silk, satin, or chiffon. Ensure the dress fits well and is comfortable for sitting and eating.

**Color :** While black is always a safe choice, don't shy away from colors that complement your skin tone and make you feel confident. Deep jewel tones like emerald green, sapphire blue, or classic red can be stunning.

**Accessories :** Pair your dress with elegant accessories such as statement earrings, a delicate necklace, or a classy clutch. Don't forget a pair of comfortable yet stylish heels.

**Details :** Look for dresses with interesting details like lace, beading, or unique necklines to add a touch of sophistication.

### **Cas d'usage du modèle Qwen-7b, avec 1 epoch du pré-entraînement léger**

**Assistant :** I'd be happy to recommend some stylish dresses with a touch of background knowledge on various styles :

#### **1. Famous French Style**

The French style is renowned for its elegance, simplicity, and timeless appeal. Here are a few dresses that embody the chic Parisian aesthetic :

- Little Black Dress (LBD) : A staple in every French woman's wardrobe, the LBD is versatile and effortlessly elegant.

- Brand : Givenchy

- The LBD was popularized by Coco Chanel and later by Audrey Hepburn in "Breakfast at Tiffany's." It's the epitome of French sophistication.

- Wrap Dress : This dress flatters all body types and is perfect for both casual and formal dinners.

- Brand : Diane von Fürstenberg

- Invented by Diane von Fürstenberg in the 1970s, it's a symbol of effortless chic, often embraced by French women for its versatility and elegance.

#### **2. Y2K Style for a Colorful Summer**

The Y2K style brings back the fun and vibrant fashion of the late '90s and early 2000s. Think bold colors, unique cuts, and a bit of nostalgia.

- Slip Dress : Lightweight and easy to wear, perfect for a summer date.

- Brand : Realisation Par

- Slip dresses were iconic in the late '90s and early 2000s, often seen on celebrities like Kate Moss. They offer a sultry yet simple look.

- Halter Neck Dress : A popular Y2K silhouette that's perfect for a night out.

- Brand : I.AM.GIA

- Halter necks were everywhere in the Y2K era, adding a playful and flirty touch to summer fashion.

### **Cas d'usage du modèle Qwen2-7b, avec 1 epoch de l'affinage léger**



#### A.4 Conception de prompt pour NLU dans la section 6.2

**Client** : I am looking for a cute dress.

**Assistant** : Great ! What's your style ? Do you have a specific color or length in mind ? Let me know so I can recommend some cute dresses for you.

**Client** : i want find a dress for a fine dinner dating.

**Assistant** : I've got just the thing for you ! How about the "Sexiest Lace Evening Party Midi Dresses For Women's Bodycon Body" ? It's a stunning and elegant dress that's perfect for a fine dinner date. The lace detailing adds a touch of sophistication, and the bodycon style is flattering and comfortable to wear. It's a great choice for a romantic evening out. Would you like to take a look at it ?

**Client** : oh that's nice, what color is it ?

**Assistant** : The "Sexiest Lace Evening Party Midi Dresses For Women's Bodycon Body" is available in a variety of colors, including black, navy, and wine red. The black option is classic and elegant, while the navy and wine red options add a touch of sophistication. Let me know which color you prefer, and I can help you find the perfect dress in your size.

**Client** : do you have other options ? pink ones for example, or other similar dresses ?

**Assistant** : I'm sorry, but we don't have pink options for the "Sexiest Lace Evening Party Midi Dresses For Women's Bodycon Body". However, if you're looking for a similar dress in pink, I recommend the "Sexiest Lace Evening Party Midi Dresses For Women's Bodycon Body" in black, navy, or wine red. They are all elegant and sophisticated, perfect for a fine dinner date. Would you like to take a look at them ?

**Client** : do you have other dresses ?

**Assistant** : I'm sorry, but we don't have other dresses in pink. If you're looking for a similar dress in pink, I recommend the "Sexiest Lace Evening Party Midi Dresses For Women's Bodycon Body" in black, navy, or wine red. They are all elegant and sophisticated, perfect for a fine dinner date. Would you like to take a look at them ?

## A.4 Conception de prompt pour NLU dans la section 6.2

### Instruction pour le classifieur binaire

You are an agent that determines if user input requires querying a fashion product database (clothing, accessories, shoes, etc). Your task is to return a single boolean response: true if the

input needs a product search among database, else false.

RESPOND WITH ONLY True OR False.

Examples of true cases:

- "Do you have blue shirts?"
- "I want to buy Nike shoes"
- "Show me some dresses for summer vacation near the sea"
- "Looking for winter jackets"
- "Are there any red sweaters?"
- "Do you sell jeans similar like this style?"
- "Do you have similar ones like this dress?"

Examples of false cases:

- "Hello"
- "How much is this t-shirt?"
- "What's your return policy?"
- "How do I track my order?"
- "Thank you"
- "What do you think about the brand Dior?"
- "This one and that one which is better?"
- "Sort the price"
- "Can I change my address?"
- "When will my order arrive?"
- "How do I return this?"

Guidelines:

1. Return true if user is:

- Asking about specific fashion items
- Looking to buy/find fashion products
- Requesting to find similar products

2. Return false if user is:

- Sending greetings or thanks
- Asking about product availability or prices
- Sorting prices among some retrieved products
- Asking about shipping/returns policies
- Asking for fashion advices or opinions
- Discussing order status
- Making small talk

## A.5 Conception de prompt pour Oracle

Nous avons ensuite conçu des instructions qui guident le modèle pendant le fine-tuning. Chaque prompt a été soigneusement élaboré pour simuler le type de requête utilisateur ou d'interaction auquel le modèle devrait répondre. Par exemple, un prompt pourrait présenter au modèle un titre de produit et lui demander d'identifier la catégorie de produit la plus pertinente ou de prédire la prochaine action de l'utilisateur en fonction de ses données de parcours de clics précédentes.

Les prompts ont été contextualisés pour les rendre plus représentatifs des scénarios du monde réel. Cela signifiait intégrer les données sélectionnées dans des contextes d'interaction utilisateur réalistes. Par exemple, au lieu de simplement présenter un titre de produit, le prompt pourrait simuler une requête de recherche utilisateur où le modèle doit déterminer la catégorie de produit la plus probable basée sur des informations incomplètes ou implicites fournies par l'utilisateur.

Lors de l'utilisation de méthodes zero-shot, nous avons constaté que formater les prompts avec des rôles système et des rôles utilisateur améliore les sorties LLM. Concevoir de meilleurs prompts peut améliorer les performances des modèles zero-shot. Pendant le fine-tuning, nous avons constaté que le prompt conçu peut ne pas être aussi important puisque le modèle est affiné pour exhiber un certain comportement quel que soit le prompt que nous choisissons pour l'entraînement.

Nous avons défini le prompt système comme suit :

Vous êtes un assistant d'achat en ligne utile. Votre tâche est d'effectuer un alignement de comportement sur la requête donnée : {QUERY} avec clic {CLICK}, après le clic, vous avez obtenu la description du produit comme : {PRODUCT DESCRIPTION}.



# Table des matières

Résumé    i

Remerciements    iii

## I : INTRODUCTION    1

### 1 Introduction    3

- 1.1 Définition et développement des agents conversationnels multimodaux et personnalisé    3
- 1.2 Originalité du projet    4
  - 1.2.1 Contexte du projet    4
  - 1.2.2 Opportunité du projet    5
  - 1.2.3 Originalité de l'approche    6
- 1.3 Problématique : défis techniques et reconfiguration du dialogue orienté-recommandation à l'ère des *LLMs*    6
  - 1.3.1 Des données désordonnées et incomplètes    7
  - 1.3.2 Une spécialisation excessive et une évaluation coûteuse    7
  - 1.3.3 L'apport et les limites des grands modèles de langage    7
  - 1.3.4 Vers une architecture hybride : arbitrages et contributions    8
  - 1.3.5 Des exigences computationnelles croissantes    9

### 2 État de l'art et recherches liées    11

- 2.1 TAL et les agents conversationnels    11
- 2.2 Jeux de données    13
- 2.3 Architecture et modèle d'agents conversationnels    14
  - 2.3.1 Modèles neuronaux dans les agents conversationnels    15
  - 2.3.2 Architectures pour les agents conversationnels *TOD*    23
- 2.4 Système de recommandation conversationnel    31
- 2.5 Personnalisation    35
- 2.6 Dialogues multimodaux    36

<b>II</b>	<b>:</b>	<b>MÉTHODE</b>	<b>39</b>
<b>3</b>		<b>Méthode proposée</b>	<b>41</b>
3.1		Méthode proposée	41
3.2		Plan d'expérimentation	43
<b>4</b>		<b>Génération de données</b>	<b>47</b>
4.1		Obtenir des données originales	47
4.1.1		Ressources numériques dans le domaine de la mode	48
4.1.2		Collecte des données	49
4.2		Construction de la base de connaissance	56
4.2.1		Catalogue	57
4.2.2		Terminologie et taxonomie	58
4.2.3		Connaissances contextuelles en embedding	61
4.2.4		Persona utilisateur	64
4.3		Génération de données de dialogue	66
4.3.1		Différentes méthodes pour obtenir les données de dialogue : collecte ou génération	67
4.3.2		Génération des dialogues par humain	73
4.3.3		Génération de dialogues par les automates	74
4.3.4		Génération de dialogue par LLMs	85
<b>5</b>		<b>Modélisation de l'agent conversationnel</b>	<b>95</b>
5.1		Modèle de bout en bout	96
5.2		Chaîne de traitement modulaire	98
5.2.1		Chaîne de traitement modulaire par <i>LLMs</i>	99
5.2.2		Module <i>NLU</i>	101
5.2.3		Module <i>DST</i>	111
5.3		Politique du dialogue : architecture d'auto-alignement	121
5.3.1		Simulateur d'utilisateur	122
5.3.2		Générateur du contenu produit	123
5.4		Optimisation Kahneman-Tversky ( <i>KTO</i> )	125
5.4.1		Fonction de valeur de base dans <i>KTO</i>	125
5.4.2		Adaptation de <i>KTO</i> au commerce électronique	126
<b>III</b>	<b>:</b>	<b>EXPÉRIMENTATION</b>	<b>129</b>
<b>6</b>		<b>Expérimentations</b>	<b>131</b>
6.1		Dialogues	131
6.1.1		Composition du jeu de données	131
6.1.2		Évaluation et analyse des dialogues générés	133

6.1.3	Pré-entraînement léger et affinage du modèle	140
6.2	Module <i>NLU</i>	141
6.2.1	Affinage des modèles	142
6.3	Module <i>DST</i> et Récupération de produits	144
6.3.1	Récupération entre énoncé et produit	144
6.3.2	Récupération entre produits similaires	148
6.4	Simulateur du client	152
6.5	Optimisation Kahneman-Tversky	155
6.5.1	Configuration expérimentale	155
6.5.2	Résultats expérimentaux	156
IV	: PERSPECTIVES	159
7	Discussion et conclusion	161
7.1	Discussion	161
7.2	Limitations et perspectives	163
7.3	Conclusion générale	165
Bibliographie 169		
A	Annexe	199
A.1	Résumé des ensembles de données	199
A.2	Évaluation par annotateurs dans la section 6.1.2	204
A.3	Démonstration et étude de cas dans la section 6.1.3	205
A.4	Conception de prompt pour <i>NLU</i> dans la section 6.2	207
A.5	Conception de prompt pour Oracle	209



Manying Zhang

## Agents conversationnels personnalisés et multi-modaux dans le domaine de la mode

### Résumé

La thèse porte sur la conception d'un agent conversationnel personnalisé et multimodal pour le commerce électronique dans la mode. L'objectif est de construire un système capable de comprendre des requêtes complexes, recommander des produits pertinents et répondre naturellement, en intégrant texte, image et caractéristiques des utilisateurs. La méthodologie repose sur trois axes. D'abord, une stratégie de collecte et de génération de données a été mise en place et un jeu de données conversationnel a été constitué, avec une base de connaissances incluant des informations sur les produits détaillées, les profils des utilisateurs et les interactions complexes. Ensuite, nous avons développé un système modulaire combinant classifieurs, modèles de plongement et grands modèles de langage, orchestrés dans une chaîne de traitement avec modules spécialisés pour la compréhension, la récupération, la génération de réponses et leur personnalisation. Enfin, nous avons exploré un mécanisme original d'auto-alignement itératif, combinant deux simulateurs : un simulateur d'utilisateur qui modélise les comportements d'achat et un simulateur de vendeur qui génère des descriptions de produits. En confrontant ces deux pôles, le système apprend à produire des descriptions adaptées à l'utilisateur. Les expériences menées montrent que ce système permet une interaction plus personnalisée et pertinente, tout en mettant en lumière les défis méthodologiques liés à la mise en œuvre d'un système de dialogue, à sa personnalisation et à l'intégration de la multimodalité. Ce travail ouvre des perspectives sur le développement d'agents conversationnels dans le cadre d'applications commerciales réelles.

**Mots-clés :** TAL, Agent conversationnel, Intelligence artificielle, Grand modèle de langue

### Personalized and Multimodal Conversational Agents in the Fashion Domain

### Abstract

In this thesis, we investigate the design of a personalized, multimodal conversational agent for e-commerce in the fashion domain. The objective is to build a system capable of understanding complex queries, recommending relevant products, and responding naturally while integrating text, images, and user-specific features. Our methodology is structured around three main components. First, we developed a rigorous data collection and generation strategy, constructing a conversational dataset and a comprehensive knowledge base containing detailed product information, user profiles, and complex interaction patterns. Second, we designed a modular system that combines classifiers, embedding models, and large language models, all orchestrated within a processing pipeline featuring specialized modules for understanding, information retrieval, response generation, and personalization. Third, we explored an original iterative self-alignment mechanism involving two simulators : a user simulator that models purchasing behaviors, and a seller simulator that generates product descriptions. By confronting these two components, the system learns to produce descriptions tailored to the user. The experiments demonstrate that this architecture supports more personalized and relevant interactions, while also revealing methodological challenges related to dialogue system implementation, personalization, and multimodal integration. Overall, this work opens promising perspectives for the development of conversational agents in real-world commercial applications.

**Keywords :** NLP, Dialogue System, Artificial Intelligence, Large Language Model