Classer, Ne Pas Générer : Classement d'Énoncés pour la Recommendation Explicable

Ben Kabongo [1] Arthur Satouf [2] Vincent Guigue [3]
(1) Sorbonne University, CNRS, ISIR, Paris, France
(2) Ills & airLiquide & Université Paris-Saclay, Montreal, Canada
(3) AgroParisTech, UMR MIA Paris-Saclay, Palaiseau, France
ben.kabongo@sorbonne-universite.fr, arthur.satouf@gmail.com,
vincent.guigue@agroparistech.fr

RÉSUMÉ

Les explications textuelles générées par des LLMs sont de plus en plus utilisées pour justifier les recommandations, mais leur évaluation reste un défi majeur. Nous plaidons pour un changement d'objectif : *classer, ne pas générer*. Nous formulons la recommandation explicable comme un problème de classement d'énoncés : le système ordonne des énoncés explicatifs candidats extraits d'avis et renvoie les top-k comme explication. Cette approche réduit les hallucinations, permet une analyse factuelle fine et favorise une évaluation standardisée et reproductible via des métriques de classement. Une évaluation fiable exige toutefois des énoncés *explicatifs*, *atomiques* et *uniques*, difficiles à obtenir à partir d'avis bruités. Nous relevons ce défi avec (i) une extraction basée sur des LLMs, et (ii) un clustering sémantique scalable regroupant les paraphrases. Nous introduisons STAR, un benchmark de classement d'énoncés, et évaluons baselines de popularité et modèles de l'état de l'art, révélant des limites fortes de la personnalisation.

ABSTRACT

Rank, Don't Generate : Statement-level Ranking for Explainable Recommendation

Textual explanations, generated with large language models (LLMs), are increasingly used to justify recommendations. Yet, evaluating these explanations remains a critical challenge. We advocate a shift in objective : *rank, don't generate*. We formalize explainable recommendation as a statement-level ranking problem, where systems rank candidate explanatory statements derived from reviews and return the top-k as explanation. This formulation mitigates hallucination, enables fine-grained factual analysis, and supports standardized, reproducible evaluation with established ranking metrics. Meaningful assessment, however, requires each statement to be *explanatory*, *atomic*, and *unique*, which is challenging to obtain from noisy reviews. We address this with (i) an LLM-based extraction pipeline producing explanatory and atomic statements, and (ii) a scalable, semantic clustering method consolidating paraphrases to enforce uniqueness. Building on this pipeline, we introduce STAR, a benchmark for statement ranking in explainable recommendation. We evaluate popularity-based baselines and state-of-the-art models. Popularity baselines are highly competitive, exposing critical limitations in personalized explanation ranking.

MOTS-CLÉS : Recommandation Explicable, Classement d'Énoncés, Grands Modèles de Langue (LLMs), Extraction d'Énoncés, Clustering Sémantique.

KEYWORDS: Explainable Recommendation, Statement-level Ranking, Large Language Models (LLMs), Statement Extraction, Semantic Clustering.

1 Introduction

Les systèmes de recommandation aident les utilisateurs à naviguer dans de vastes catalogues en fournissant des suggestions personnalisées. Mais l'adoption croissante d'architectures d'apprentissage profond (He *et al.*, 2020; Ma *et al.*, 2019) les rend opaques. La recommandation explicable vise à y remédier en produisant des justifications compréhensibles, essentielles pour la transparence et la confiance. Parmi les approches existantes, les explications textuelles sont particulièrement attractives : de récents travaux (Li *et al.*, 2023b; Ma *et al.*, 2024; Xie *et al.*, 2023) s'appuient de plus en plus sur des LLMs (Dubey *et al.*, 2024; Yang *et al.*, 2025) pour générer des explications en langage naturel. Toutefois, évaluer de façon fiable la qualité de ces explications demeure un défi majeur.

Les premiers protocoles d'évaluation reposent sur des métriques lexicales (Lin, 2004; Papineni *et al.*, 2002), peu robustes aux paraphrases. Les métriques sémantiques (Yuan *et al.*, 2021; Zhang *et al.*, 2019) atténuent cette limitation, mais la similarité sémantique ne garantit pas l'ancrage factuel (Honovich *et al.*, 2022; Laban *et al.*, 2022). Les évaluations avec des LLMs (Fu *et al.*, 2023; Li *et al.*, 2024) offrent des jugements plus riches, mais les LLMs sont très sensibles au prompt et souvent propriétaires, donc difficiles à reproduire (Xu *et al.*, 2025). Côté génération, d'autres échecs apparaissent : les modèles hallucinent ou génèrent des explications génériques plutôt que spécifiques à l'interaction (Kabongo & Guigue, 2025; Li *et al.*, 2021a). Enfin, les explications étant multi-facettes, les protocoles actuels manquent d'indiquer quels facteurs sont réellement étayés et leur importance relative.

Classement d'énoncés pour la recommandation explicable. Ces limites motivent une reformulation de la recommandation explicable textuelle comme un problème de *classement* d'*énoncés* explicatifs. Dans la lignée de travaux antérieurs (Li *et al.*, 2021a, 2023a; Wei *et al.*, 2023), nous défendons un changement de paradigme : *classer, ne pas générer*. Au lieu de produire un paragraphe libre, le système classe, pour chaque interaction utilisateur–item, des énoncés candidats du plus au moins pertinent et renvoie les top-k comme explication. Cette formulation a trois avantages : (i) elle décompose l'explication en unités vérifiables ancrées dans les avis, réduisant les hallucinations et facilitant l'analyse fine de la fidélité ; (ii) elle modélise l'importance relative des facteurs via des scores de pertinence ; (iii) elle permet une évaluation standardisée et reproductible avec des métriques de classement (p. ex., Precision, Recall, NDCG), favorisant des comparaisons objectives.

Malgré son intérêt, cette reformulation pose des défis de construction de données et d'évaluation. Une évaluation pertinente exige que chaque énoncé vérifie trois propriétés : (i) *explicativité* — décrire des faits généraux sur l'item influençant l'expérience plutôt que des détails circonstanciels ou personnels sans valeur explicative ; (ii) *atomicité* — exprimer une seule opinion sur un seul aspect ; (iii) *unicité* — regrouper les déclarations sémantiquement équivalentes en une seule cible pour éviter de biaiser les métriques. Pour y répondre, EXTRA (Li *et al.*, 2021a) extrait des phrases fréquentes et déduplique les paraphrases via similarité lexicale en n-grammes. En pratique, ce regroupement purement lexical omet souvent des paraphrases sémantiquement équivalentes formulées différemment, et l'extraction depuis des avis bruts reste parasitée par du contenu non explicatif ou non atomique.

Notre contribution. Nous proposons de meilleures méthodes d'extraction et de clustering. Pour l'extraction, nous introduisons une procédure avec des LLMs en deux étapes : (i) *extraction de candidats*, qui produit des énoncés explicatifs et atomiques en filtrant le bruit des avis ; (ii) *vérification*, qui élimine les sorties non conformes. Pour le clustering, nous proposons une procédure scalable, pilotée par la sémantique : (i) *recherche de voisins proches approximatifs* via embeddings denses ; (ii) *filtrage pair-à-pair* avec un cross-encoder ; (iii) *raffinement* en construisant un graphe de similarité, en extrayant des composantes connexes puis en scindant les groupes peu cohésifs.

Sur cette base, nous introduisons STAR (**Sta**tement **R**anking), un nouveau benchmark de classement d'énoncés en recommandation explicable, construit à partir de quatre catégories d'Amazon Reviews 2014 (Ni *et al.*, 2019). Nous validons STAR avec de l'évaluation humaine et automatique, montrons les limites de EXTRA (Li *et al.*, 2021a) et l'intérêt de nos procédures. Enfin, nous introduisons des baselines de popularité et les comparons sur STAR à BPER+ (Li *et al.*, 2023a) et EXPGCN (Wei *et al.*, 2023), sous deux axes de classement : *global-level* (tous les énoncés) et *item-level* (énoncés de l'item). Si EXPGCN domine en global-level, les signaux de fréquence restent très compétitifs ; en item-level, une baseline de popularité utilisateur dépasse (+0.08 NDCG@10 sur *Toys*) ou égale EXPGCN, révélant un fort déficit de personnalisation et le besoin de protocoles reproductibles. Nous fournissons notre code et les jeux de données pour supporter la reproductibilité et les travaux futurs. [1]

2 Travaux Connexes

Recommandation explicable textuelle. Les explications textuelles ont d'abord été produites via des *templates* (Wang *et al.*, 2018; Zhang *et al.*, 2014), mais ces approches manquent de flexibilité et de diversité. Les travaux suivants ont généré directement l'avis de l'utilisateur comme explication (Dong *et al.*, 2017; Kabongo *et al.*, 2025a,b; Li *et al.*, 2017, 2021b, 2023b; Xie *et al.*, 2023). Comme les avis contiennent souvent du bruit et des passages non explicatifs, des méthodes récentes (Ma *et al.*, 2024; Li *et al.*, 2025) les transforment en paragraphes explicitement justificatifs, en s'appuyant sur des LLMs (Dubey *et al.*, 2024; Yang *et al.*, 2025). Toutefois, l'évaluation fiable de la qualité explicative reste un défi central.

Évaluation des explications textuelles. Mesurer automatiquement la qualité d'un texte généré reste difficile. Les métriques lexicales en n-grammes (BLEU, ROUGE) (Papineni *et al.*, 2002; Lin, 2004) sont peu robustes aux paraphrases. Les métriques sémantiques, à base d'embeddings ou apprises (Zhang *et al.*, 2019; Sellam *et al.*, 2020; Yuan *et al.*, 2021), gèrent mieux les reformulations, mais la similarité sémantique ne garantit pas la cohérence factuelle avec les évidences (Honovich *et al.*, 2022; Laban *et al.*, 2022). Les LLMs comme évaluateurs (Fu *et al.*, 2023; Liu *et al.*, 2023) peuvent mieux s'aligner sur l'humain, mais sont souvent propriétaires, sensibles au prompt et difficiles à reproduire (Sheng *et al.*, 2025; Xu *et al.*, 2025). Enfin, les modèles peuvent produire des explications génériques (Li *et al.*, 2021a) ou halluciner (Kabongo & Guigue, 2025), tandis que l'évaluation au niveau paragraphe reste peu diagnostique et n'isole ni les facteurs étayés ni leur importance relative.

Benchmarks et méthodes de classement d'explications. EXTRA (Li *et al.*, 2021a) propose un benchmark qui standardise l'évaluation via des métriques de classement. Il construit des candidats à partir d'avis et met en évidence deux défis : filtrer le bruit non explicatif et regrouper les paraphrases. Son extraction, fondée sur des heuristiques (p. ex., phrases récurrentes, filtrage de pronoms), peut supprimer des explications utiles ou conserver de l'irrélevant, tandis que son clustering par LSH en n-grammes (Anand & Jeffrey David, 2011) capture mal la similarité sémantique au-delà du recouvrement lexical. Côté modèles, BPER+ (Li *et al.*, 2023a) factorise la relation utilisateur–item–énoncé et s'appuie sur BERT (Devlin *et al.*, 2019), et EXPGCN (Wei *et al.*, 2023) apprend des représentations via convolutions de graphe. Cependant, ces travaux évaluent essentiellement le *global-level*. Nous introduisons des baselines de popularité et étendons l'évaluation au *item-level*, montrant que les signaux de popularité sont très forts pour le classement d'explications personnalisées.

1. https://github.com/BenKabongo25/Statement_Ranking_Explainable_Recommendation

3 Classement d'énoncés pour la recommandation explicable

Le classement d'énoncés en recommandation explicable consiste à ordonner des phrases explicatives candidates (*énoncés*) selon leur capacité à justifier une interaction utilisateur–item, puis à retourner les top-k comme explication.

3.1 Propriétés des énoncés

Pour une évaluation fiable et reproductible, les énoncés doivent respecter trois propriétés : (i) *Explicativité* : l'énoncé doit exprimer un fait ou une opinion générale sur l'item influençant l'expérience, et exclure les détails circonstanciels ou personnels. (ii) *Atomicité* : l'énoncé doit porter sur une seule opinion et un seul aspect. Une phrase combinant plusieurs aspects rend l'appariement et la vérification ambigus. (iii) *Unicité* : les paraphrases doivent être regroupées en une représentation canonique, afin d'éviter que des quasi-doublons ne biaisent les métriques.

3.2 Formulation du problème

Soient $\mathcal{U}$ l'ensemble des utilisateurs et $\mathcal{I}$ celui des items. À chaque interaction (u, i) est associé un ensemble d'énoncés explicatifs $\mathcal{S}_{ui}$ (vérité terrain), éventuellement annotés par une polarité $p_s \in \{\text{POS}, \text{NEG}, \text{NEU}\}$ pour chaque énoncé s. On note $\mathcal{S}_i = \bigcup_u \mathcal{S}_{ui}$ l'ensemble des énoncés liés à l'item i, et $\mathcal{S} = \bigcup_{u,i} \mathcal{S}_{ui}$ l'univers global. Comme les annotations en énoncés explicatifs sont rares, nous décrivons ensuite notre approche d'extraction automatique à partir des avis utilisateurs.

Le classement d'énoncés pour expliquer l'interaction (u, i) consiste à prédire un score de pertinence $\hat{r}_{uis}$ pour chaque énoncé candidat s, à les ordonner par score décroissant, puis à retourner les k premiers comme explication. L'ensemble vérité terrain $\mathcal{S}_{ui}$ est inclus dans l'ensemble des candidats considérés. On distingue deux cadres d'évaluation. (i) *Global-level* : les candidats sont tirés de l'univers global $\mathcal{S}$ (tous les énoncés du dataset), ce qui peut conduire à des explications moins spécifiques à l'item. (ii) *Item-level* : les candidats sont restreints à $\mathcal{S}_i$ (énoncés associés à l'item i), garantissant des explications item-spécifiques.

3.3 Évaluation

Pour une interaction (u, i), on note $\mathcal{S}_{ui}$ l'ensemble des énoncés explicatifs de vérité terrain, et $\pi_{ui}(j)$ l'énoncé retourné au rang j dans la liste classée par ordre de pertinence ($\mathcal{S}$ pour *global-level* et $\mathcal{S}_i$ pour *item-level*). Nous évaluons la qualité du classement avec des métriques standard de recherche d'information : Precision (P@k), Recall (R@k) et NDCG@k. Les métriques s'écrivent alors :

$$\mathrm{rel}_j \;=\; \delta\big(\pi_{ui}(j) \in \mathcal{S}_{ui}\big) \quad \mathrm{P@}k(u,i) = \frac{1}{k}\sum_{j=1}^{k} \mathrm{rel}_j \quad \mathrm{R@}k(u,i) = \frac{1}{|\mathcal{S}_{ui}|}\sum_{j=1}^{k} \mathrm{rel}_j$$

$$\mathrm{NDCG@}k(u,i) = \frac{1}{Z_k}\sum_{j=1}^{k} \frac{2^{\mathrm{rel}_j}-1}{\log_2(j+1)} \quad Z_k = \sum_{j=1}^{k} \frac{1}{\log_2(j+1)}, \qquad (1)$$

où $\delta(\cdot)$ est la fonction indicatrice, et rel_j la pertinence binaire au rang j.

4 Benchmark STAR

Nous présentons STAR, un benchmark de *classement d'énoncés* pour la recommandation explicable. STAR est construit à partir d'une extraction d'énoncés via LLM pour garantir *explicativité* et *atomicité*, et d'un clustering sémantique à grande échelle pour imposer l'*unicité* en consolidant les paraphrases, sur quatre catégories d'Amazon Reviews 2014 (Ni *et al.*, 2019). Nous présentons dans cette section une brève description de notre approche, et fournissons plus de détails en annexe.

4.1 Extraction et vérification des énoncés

Les avis contiennent des preuves explicatives, mais aussi du bruit. Nous utilisons une procédure en deux étapes (Fig. 1) : (i) *extraction de candidats*, où un LLM extrait des énoncés (avec leur polarité) depuis l'avis de l'utilisateur u pour l'item i ; (ii) *vérification*, où un second LLM filtre les sorties non explicatives, non atomiques ou redondantes pour produire $\mathcal{S}_{ui}$.

Nous utilisons `Qwen3-14B` (Yang *et al.*, 2025) pour l'extraction, `Qwen3-8B` (Yang *et al.*, 2025) pour la vérification, et fournissons les prompts finaux en annexe.

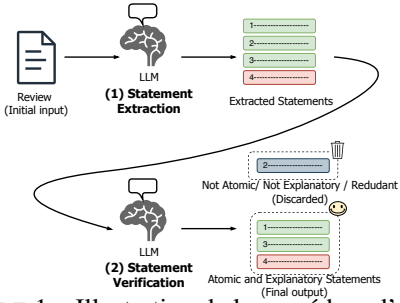


FIGURE 1 – Illustration de la procédure d'extraction et de vérification des énoncés (explicatifs et atomiques) avec des LLMs à partir des avis.

4.2 Clustering des paraphrases

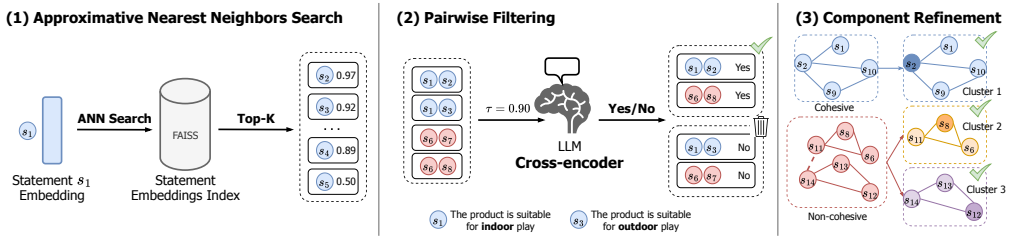


FIGURE 2 – Illustration de la procédure de clustering des énoncés paraphrases.

Pour imposer l'*unicité*, nous regroupons les paraphrases via un pipeline sémantique en trois étapes (Fig. 2) : (i) *recherche approximative de voisins* (ANN) avec le modèle d'embeddings denses `Qwen3-Embedding-0.6B` (Zhang *et al.*, 2025) pour retrouver, pour chaque énoncé, les K énoncés les plus proches par polarité ; (ii) *filtrage par paires* avec un cross-encoder `Qwen3-Reranker-0.6B` (Zhang *et al.*, 2025) pour ne retenir que les paires de paraphrases avérées ; (iii) *raffinement* par composantes connexes du graphe de similarité, avec découpage des composantes peu cohésives. Le représentant de chaque cluster est l'énoncé le plus central, formant l'ensemble canonique $\mathcal{S}$.

4.3 Jeux de données STAR

Nous appliquons ce pipeline à quatre catégories d'Amazon Reviews 2014 (Ni *et al.*, 2019) : *Toys*, *Clothes*, *Beauty* et *Sports*. Nous conservons les interactions ayant au moins un énoncé, obtenant 115K–294K interactions et 718K–1.3M triplets utilisateur–item–énoncé après clustering. Nous effectuons un split temporel par utilisateur : dernière interaction pour le test, avant-dernière pour la validation, le reste pour l'entraînement. Les statistiques complètes du benchmark figurent dans la Table 6.

5 Expérimentations

5.1 Classement d'énoncés sur STAR

5.1.1 Méthodes

Nous évaluons six méthodes de classement d'énoncés sur STAR : la baseline RANDOM, trois baselines de popularité (USERPOP, ITEMPOP, GLOBALPOP), et deux méthodes de l'état de l'art pour le classement de passages explicatifs (BPER+ (Li *et al.*, 2023a), EXPGCN (Wei *et al.*, 2023)).

Baselines de popularité. Nous introduisons trois baselines fondées sur la fréquence des énoncés dans les données d'entraînement :

— *User-based Popularity* (USERPOP) : fréquence dans l'historique de l'utilisateur :

$$\hat{r}_{uis} = \sum_{i' \in \mathcal{I}_u} \delta(s \in \mathcal{S}_{ui'}) \tag{2}$$

— *Item-based Popularity* (ITEMPOP) : fréquence dans les interactions de l'item :

$$\hat{r}_{uis} = \sum_{u' \in \mathcal{U}_i} \delta(s \in \mathcal{S}_{u'i}) \tag{3}$$

— *Global Popularity* (GLOBALPOP) : fréquence globale sur toutes les interactions :

$$\hat{r}_{uis} = \sum_{i' \in \mathcal{I}} \sum_{u' \in \mathcal{U}_{i'}} \delta(s \in \mathcal{S}_{u'i'}) \tag{4}$$

Méthodes de l'état de l'art. BPER+ décompose la relation ternaire utilisateur–item–énoncé en interactions binaires, enrichit les représentations d'énoncés avec BERT (Devlin *et al.*, 2019), et combine les scores utilisateur et item via un poids μ. EXPGCN apprend des représentations par convolutions sur graphe et calcule la pertinence en sommant les scores utilisateur et item.

5.1.2 Protocole expérimental

Nous réutilisons les implémentations officielles de BPER+ [2] et EXPGCN [3], et implémentons les baselines. Pour BPER+, nous utilisons `Bert-base-uncased` pour les embeddings d'énoncés,

2. https://github.com/lileipisces/BPER
3. https://github.com/Joinn99/ExpGCN

Methodes	Global-level			Item-level		
	P@10	R@10	N@10	P@10	R@10	N@10
Toys						
RANDOM	0.00119	0.00285	0.00439	0.07594	0.18552	0.12951
USERPOP	0.03814	0.09697	0.07890	**0.13901**	**0.32613**	**0.26999**
ITEMPOP	0.04292	0.10473	0.09410	0.04680	0.11625	0.09928
GLOBALPOP	0.03305	0.08487	0.07172	0.09357	0.24662	0.18156
BPER+	0.03454	0.08893	0.07984	0.09417	0.24909	0.18545
Gain	-0.00839	-0.01580	-0.01426	-0.04485	-0.07705	-0.08454
EXPGCN	**0.04509****	**0.11686*****	**0.10054*****	0.09279*****	0.24393*****	0.18618*****
Gain	0.00216	0.01213	0.00643	-0.04622	-0.08220	-0.08381
Clothes						
RANDOM	0.00002	0.00005	0.00005	0.09123	0.22660	0.15632
USERPOP	0.03416	0.09436	0.08464	0.12156	0.31049	0.25325
ITEMPOP	0.04646	0.12454	0.12000	0.05376	0.14419	0.12947
GLOBALPOP	0.05905	0.15790	0.14494	**0.12225**	**0.32373**	0.25389
BPER+	0.05246	0.14202	0.10118	0.12173	0.32321	0.24685
Gain	-0.00659	-0.01589	-0.04376	-0.00052	-0.00052	-0.00705
EXPGCN	**0.06469*****	**0.17267*****	**0.15976*****	0.12164	0.32241	**0.25646**
Gain	0.00564	0.01477	0.01481	-0.00061	-0.00133	0.00256
Beauty						
RANDOM	0.00098	0.00264	0.00392	0.06682	0.16872	0.11619
USERPOP	0.01885	0.05036	0.04358	0.09054	0.23200	**0.18171**
ITEMPOP	0.04099	0.10350	0.09264	0.04468	0.11374	0.09731
GLOBALPOP	0.02596	0.06886	0.05478	0.08687	0.22809	0.16637
BPER+	0.03179	0.08327	0.06910	0.08840	0.23327	0.17112
Gain	-0.00920	-0.02022	-0.02353	-0.00214	0.00126	-0.01059
EXPGCN	**0.04564*****	**0.11808*****	**0.10001*****	**0.09056**	**0.23750***	0.18159
Gain	0.00464	0.01458	0.00737	0.00002	0.00550	-0.00011
Sports						
RANDOM	0.00000	0.00000	0.00001	0.06899	0.15810	0.11322
USERPOP	0.01697	0.04250	0.03434	**0.08776**	**0.20677**	**0.16326**
ITEMPOP	0.03626	0.08859	0.07921	0.03893	0.09542	0.08252
GLOBALPOP	0.03075	0.07589	0.06309	0.08481	0.20418	0.15370
BPER+	0.01529	0.03939	0.02486	0.08282	0.20070	0.14371
Gain	-0.02098	-0.04920	-0.05434	-0.00494	-0.00606	-0.01955
EXPGCN	**0.04055*****	**0.09852*****	**0.08639*****	0.08373*****	0.20148	0.15978*
Gain	0.00429	0.00994	0.00718	-0.00403	-0.00529	-0.00348

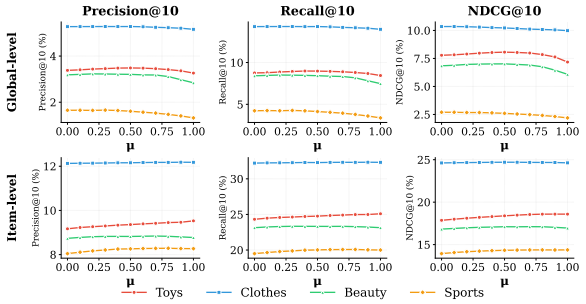
TABLE 1 – Résultats du classement d'énoncés sur STAR. Nous reportons les métriques P/R/NDCG à $k = 10$ pour le niveau global et le niveau item. Le *gain* (pour BPER+ et EXPGCN) est le gain relatif par rapport à la meilleure baseline de popularité par métrique. Niveaux de signification (test-t par rapport à la meilleure baseline) : * $p < 0,05$, ** $p < 0,01$, *** $p < 0,001$.

avec $d = 20$, $\gamma = 10^{-5}$ et $\mu = 0.7$ après tuning. Pour EXPGCN, nous fixons $d = 128$, $\gamma = 10^{-3}$, et $L = 4$ couches pour les sous-graphes utilisateur–énoncé et item–énoncé, et $L = 2$ pour le sous-graphe utilisateur–item. L'entraînement est limité à 500 époques ; nous rapportons la moyenne sur 3 runs, en sélectionnant le meilleur modèle sur la validation par NDCG@10. Nous analysons également la sensibilité des modèles à leurs hyperparamètres clés (μ pour BPER+, L pour EXPGCN).

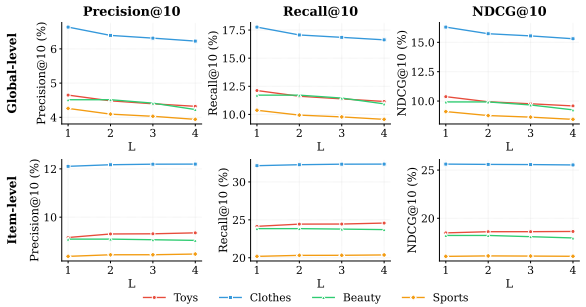
5.1.3 Résultats

Global-level. La Table 1 (Global-level) présente les résultats en classement global-level. EXPGCN obtient les meilleures performances sur tous les jeux de données et toutes les métriques, avec un gain de NDCG@10 sur la meilleure baseline allant de $+0.00643$ sur *Toys* jusqu'à $+0.01481$ sur *Clothes*. Les baselines de popularité restent très compétitives : ITEMPOP est généralement la plus forte, sauf sur *Clothes* où GLOBALPOP domine, et GLOBALPOP surpasse systématiquement USERPOP. En revanche, BPER+ est en deçà de la meilleure baseline de popularité sur tous les jeux de données, avec la plus forte dégradation sur *Sports* (-0.05434 en NDCG@10), suggérant qu'un équilibre imparfait entre signaux utilisateur et item peut fortement nuire à la qualité du classement dans ce cadre. En global-level, les signaux de fréquence item et globaux dominent, et la modélisation par graphe de la structure utilisateur–item–énoncé permet des gains constants par rapport à ces priors fréquentiels.

Item-level. La Table 1 (Item-level) révèle une dynamique différente. USERPOP devient une baseline particulièrement forte sur *Sports* et, plus nettement encore, sur *Toys* où EXPGCN lui est significativement inférieur sur toutes les métriques (-0.08381 en NDCG@10). À l'inverse, ITEMPOP s'effondre dans ce cadre et est surpassé par RANDOM sur tous les jeux de données : une fois l'item fixé, la fréquence item ne constitue plus un signal discriminant. La supériorité de USERPOP reste toutefois dépendante du jeu de données : sur *Clothes*, les meilleurs résultats sont partagés entre GLOBALPOP et EXPGCN selon la métrique. BPER+ est généralement surpassé par USERPOP. Ces résultats mettent en évidence un déficit de personnalisation : les modèles performants en global-level ne traduisent pas systématiquement leurs gains en re-classement personnalisé au niveau item, tandis que la fréquence dans l'historique utilisateur constitue un signal robuste et difficile à dépasser.



(a) Impact du paramètre μ sur les performances de BPER+.



(b) Impact du paramètre L sur les performances de EXPGCN.

FIGURE 3 – Sensibilité aux hyperparamètres.

Sensibilité aux hyperparamètres. La Figure 3a illustre l'effet de μ sur BPER+. En global-level, les performances sont optimales pour des valeurs intermédiaires de μ, et se dégradent lorsque μ tend vers 1,

indiquant qu'un équilibre entre signaux utilisateur et item est nécessaire. En item-level, des valeurs élevées de μ améliorent monotonement les performances, confirmant l'importance du signal utilisateur dans ce cadre. La Figure 3b montre la sensibilité de ExpGCN au nombre de couches L. En global-level, augmenter L dégrade systématiquement les performances, indiquant qu'une propagation trop profonde est nuisible. En item-level, l'effet est plus faible, avec des variations marginales selon les jeux de données.

5.2 Évaluation de la qualité du benchmark StaR

5.2.1 Qualité de l'extraction

Nous évaluons l'explicativité et l'atomicité des énoncés extraits par évaluation humaine sur 75 interactions par dataset et automatique avec `Llama-3.1-8B-Instruct` (Dubey *et al.*, 2024) sur 10 000 interactions par dataset, et reportons les résultats dans la Table 2. Avec l'étape de vérification, l'évaluation humaine indique 92–96% d'énoncés explicatifs et 81–94% d'énoncés atomiques ; l'évaluation automatique donne des estimations comparables (93–97% et 91–96% respectivement). L'étape de vérification améliore systématiquement la qualité, avec des gains allant jusqu'à plus de 10%. La Table 4 illustre les limites de la sélection heuristique des énoncés de Extra face à StaR qui exploite la compréhension du langage naturel des LLMs.

	Explicativité			Atomicité		
Vérification	✗	✓	*Gain*	✗	✓	*Gain*
Évaluation Humaine						
Toys	79.71	93.93	+14.22	71.53	83.75	+12.22
Clothes	83.46	95.65	+12.19	86.35	93.87	+7.52
Beauty	84.09	92.16	+8.07	66.75	80.92	+14.17
Sports	85.12	96.09	+10.97	79.97	88.48	+8.51
Évaluation Automatique (LLM)						
Toys	82.70	93.41	+10.71	80.37	91.36	10.99
Clothes	86.66	94.62	+7.96	84.16	92.86	+8.7
Beauty	82.66	93.51	+10.85	80.30	91.40	+11.1
Sports	92.63	97.20	+4.57	90.21	95.50	+5.29

TABLE 2 – Évaluation de la qualité (*explicativité* et *atomicité*) des énoncés sur le benchmark StaR.

5.2.2 Qualité du clustering

Nous évaluons l'unicité via la dispersion intra-cluster (SSE) et la séparation inter-cluster (SSB) (Palacio-Niño & Berzal, 2019), calculées avec le modèle d'embedding `KaLM-Embedding-Gemma3-12B-2511` (Zhao *et al.*, 2025). Nous reportons les résultats dans la Table 3. Notre pipeline complet réduit le nombre d'énoncés de 40–55% et obtient le meilleur SSB, tandis que le clustering de Extra produit un regroupement quasi trivial ($\leq$0.22% de réduction), incapable de fusionner les paraphrases au-delà de la similarité lexicale. Nous présentons quelques exemples qualitatifs de clustering avec notre méthode dans le Table 5.

Dataset	Methode	#Clust.	Red.%	SSE ↓	SSB ↑
Toys	Extra	472 019	0.17	**0.0000**	0.0785
	StaR	281 664	40.43	0.0033	**0.0818**
Clothes	Extra	569 103	0.22	**0.0000**	0.0713
	StaR	260 477	54.33	0.0068	**0.0748**
Beauty	Extra	506 869	0.16	**0.0000**	0.0725
	StaR	229 448	54.80	0.0065	**0.0760**
Sports	Extra	942 442	0.14	**0.0000**	0.0810
	StaR	556 209	40.98	0.0039	**0.0836**

TABLE 3 – Évaluation de la qualité du clustering des énoncés. Nous présentons des statistiques de réduction et des mesures non supervisées (SSE, SSB).

Avis	EXTRA	STAR (énoncé, sentiment)
My grand daughter has had so much fun with this set. She plays for hours and enjoys every minute. Good Job ! Great quality toy.	*good job*	the product is enjoyable to use (POS) ; the product is of great quality (POS) ; the product is a toy (NEU)
Bought for my 5 year old nephew. Great toy ! ! He loves it ! Stickers were great and easy to open the mystery machine.	*great toy*	the stickers look great (POS) ; the product is easy to open (POS) ; the product is a toy (NEU)

TABLE 4 – Comparaison qualitative de l'extraction d'énoncés explicatifs sur *Toys* entre EXTRA et STAR.

Cluster 1 : Preschoolers **Repr.** the product is good for preschoolers **Members** the product is perfect for preschoolers ; the product is suitable for preschool ; the product is good for preschool age
Cluster 2 : Breaks after use **Repr.** the product breaks after some use **Members** the product breaks after a few uses ; the product broke as soon as it is used ; the product fell apart after a few months of use
Cluster 3 : Encourages imagination **Repr.** the product encourages a child's imagination **Members** the product encourages children's imagination ; the product encourages imaginative thinking ; the product encourages a child to use their imagination

TABLE 5 – Exemples qualitatifs du clustering d'énoncés sur *Toys* de STAR. Chaque exemple montre une énoncé représentatif et quelques membres du même cluster.

6 Conclusion

Dans cet article, nous défendons un changement d'objectif pour la recommandation explicable textuelle : *classer, plutôt que générer*. En formulant l'explication comme un *classement d'énoncés* parmi des candidats extraits d'avis, cette approche mitige les hallucinations par construction et permet une évaluation standardisée et reproductible à l'aide de métriques de classement établies. Pour rendre cette évaluation significative, nous avons défini trois propriétés clés pour les énoncés (*explicativité* : faits sur l'item affectant l'expérience utilisateur ; *atomicité* : une opinion sur un aspect ; *unicité* : absence de paraphrases redondantes), via un pipeline LLM d'extraction et de vérification en deux étapes, complété par un clustering sémantique à grande échelle pour consolider les paraphrases au-delà de la similarité lexicale. En nous appuyant sur ce pipeline, nous avons introduit STAR, un benchmark couvrant quatre catégories d'Amazon Reviews 2014, et évalué le classement au niveau global (tous les énoncés du jeu de données) et au niveau item (énoncés propres à l'item cible). Nos résultats révèlent une forte asymétrie entre ces deux cadres : les modèles de l'état de l'art sont compétitifs en global-level aux côtés de priors fréquentiels forts, mais en item-level, une simple baseline de popularité fondée sur l'historique utilisateur peut les égaler ou les surpasser, exposant un déficit de personnalisation persistant pour le classement fidèle des explications.

Références

ACHIAM J., ADLER S., AGARWAL S., AHMAD L., AKKAYA I., ALEMAN F. L., ALMEIDA D., ALTENSCHMIDT J., ALTMAN S., ANADKAT S. *et al.* (2023). Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*.

ANAND R. & JEFFREY DAVID U. (2011). *Mining of massive datasets*. Cambridge university press.

BAI J., BAI S., CHU Y., CUI Z., DANG K., DENG X., FAN Y., GE W., HAN Y., HUANG F. *et al.* (2023). Qwen technical report. *arXiv preprint arXiv :2309.16609*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics : human language technologies, volume 1 (long and short papers)*, p. 4171–4186.

DONG L., HUANG S., WEI F., LAPATA M., ZHOU M. & XU K. (2017). Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 623–632.

DOUZE M., GUZHVA A., DENG C., JOHNSON J., SZILVASY G., MAZARÉ P.-E., LOMELI M., HOSSEINI L. & JÉGOU H. (2025). The faiss library. *IEEE Transactions on Big Data*.

DUBEY A., JAUHRI A., PANDEY A., KADIAN A., AL-DAHLE A., LETMAN A., MATHUR A., SCHELTEN A., YANG A., FAN A. *et al.* (2024). The llama 3 herd of models. *arXiv e-prints*, p. arXiv–2407.

FU J., NG S.-K., JIANG Z. & LIU P. (2023). Gptscore : Evaluate as you desire. *arXiv preprint arXiv :2302.04166*.

HE X., DENG K., WANG X., LI Y., ZHANG Y. & WANG M. (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.

HONOVICH O., AHARONI R., HERZIG J., TAITELBAUM H., KUKLIANSY D., COHEN V., SCIALOM T., SZPEKTOR I., HASSIDIM A. & MATIAS Y. (2022). True : Re-evaluating factual consistency evaluation. *arXiv preprint arXiv :2204.04991*.

KABONGO B. & GUIGUE V. (2025). On the factual consistency of text-based explainable recommendation models. *arXiv preprint arXiv :2512.24366*.

KABONGO B., GUIGUE V. & LEMBERGER P. (2025a). Elixir : Efficient and lightweight model for explaining recommendations. *arXiv preprint arXiv :2508.20312*.

KABONGO B., GUIGUE V. & LEMBERGER P. (2025b). Prédiction des préférences et génération de revue personnalisée basées sur les aspects et attention. In *Actes de la 20e Conférence en Recherche d'Information et Applications (CORIA)*, p. 151–170.

LABAN P., SCHNABEL T., BENNETT P. N. & HEARST M. A. (2022). Summac : Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, **10**, 163–177.

LI H., DONG Q., CHEN J., SU H., ZHOU Y., AI Q., YE Z. & LIU Y. (2024). Llms-as-judges : a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv :2412.05579*.

LI L., ZHANG Y. & CHEN L. (2021a). Extra : Explanation ranking datasets for explainable recommendation. In *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 2463–2469.

LI L., ZHANG Y. & CHEN L. (2021b). Personalized transformer for explainable recommendation. *arXiv preprint arXiv :2105.11601*.

LI L., ZHANG Y. & CHEN L. (2023a). On the relationship between explanation and recommendation : Learning to rank explanations for improved performance. *ACM Transactions on Intelligent Systems and Technology*, **14**(2), 1–24.

LI L., ZHANG Y. & CHEN L. (2023b). Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, **41**(4), 1–26.

LI P., WANG Z., REN Z., BING L. & LAM W. (2017). Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 345–354.

LI Y., ZHANG X., LUO L., CHANG H., REN Y., KING I. & LI J. (2025). G-refer : Graph retrieval-augmented large language model for explainable recommendation. In *Proceedings of the ACM on Web Conference 2025*, p. 240–251.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out*, p. 74–81.

LIU Y., ITER D., XU Y., WANG S., XU R. & ZHU C. (2023). G-eval : Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv :2303.16634*.

MA J., ZHOU C., CUI P., YANG H. & ZHU W. (2019). Learning disentangled representations for recommendation. *Advances in neural information processing systems*, **32**.

MA Q., REN X. & HUANG C. (2024). Xrec : Large language models for explainable recommendation. *arXiv preprint arXiv :2406.02377*.

NI J., LI J. & MCAULEY J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, p. 188–197.

PALACIO-NIÑO J.-O. & BERZAL F. (2019). Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv :1905.05667*.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, p. 311–318.

SELLAM T., DAS D. & PARIKH A. P. (2020). Bleurt : Learning robust metrics for text generation. *arXiv preprint arXiv :2004.04696*.

SHENG H., LIU X., HE H., ZHAO J. & KANG J. (2025). Analyzing uncertainty of llm-as-a-judge : Interval evaluations with conformal prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, p. 11297–11339.

SUN W., YAN L., MA X., WANG S., REN P., CHEN Z., YIN D. & REN Z. (2023). Is chatgpt good at search ? investigating large language models as re-ranking agents. *arXiv preprint arXiv :2304.09542*.

WANG N., WANG H., JIA Y. & YIN Y. (2018). Explainable recommendation via multi-task learning in opinionated text data. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, p. 165–174.

WANG S., SUN X., LI X., OUYANG R., WU F., ZHANG T., LI J., WANG G. & GUO C. (2025). Gpt-ner : Named entity recognition via large language models. In *Findings of the association for computational linguistics : NAACL 2025*, p. 4257–4275.

WANG Z., XIE Q., FENG Y., DING Z., YANG Z. & XIA R. (2023). Is chatgpt a good sentiment analyzer ? a preliminary study. *arXiv preprint arXiv :2304.04339*.

WEI T., CHOW T. W., MA J. & ZHAO M. (2023). Expgcn : Review-aware graph convolution network for explainable recommendation. *Neural Networks*, **157**, 202–215.

XIE Z., SINGH S., MCAULEY J. & MAJUMDER B. P. (2023). Factual and informative review generation for explainable recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, p. 13816–13824.

XU Y., RUIS L., ROCKTÄSCHEL T. & KIRK R. (2025). Investigating non-transitivity in llm-as-a-judge. *arXiv preprint arXiv :2502.14074*.

YANG A., LI A., YANG B., ZHANG B., HUI B., ZHENG B., YU B., GAO C., HUANG C., LV C. *et al.* (2025). Qwen3 technical report. *arXiv preprint arXiv :2505.09388*.

YUAN W., NEUBIG G. & LIU P. (2021). Bartscore : Evaluating generated text as text generation. *Advances in neural information processing systems*, **34**, 27263–27277.

ZHANG T., KISHORE V., WU F., WEINBERGER K. Q. & ARTZI Y. (2019). Bertscore : Evaluating text generation with bert. *arXiv preprint arXiv :1904.09675*.

ZHANG T., LADHAK F., DURMUS E., LIANG P., MCKEOWN K. & HASHIMOTO T. B. (2024). Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, **12**, 39–57.

ZHANG Y., LAI G., ZHANG M., ZHANG Y., LIU Y. & MA S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, p. 83–92.

ZHANG Y., LI M., LONG D., ZHANG X., LIN H., YANG B., XIE P., YANG A., LIU D., LIN J. *et al.* (2025). Qwen3 embedding : Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv :2506.05176*.

ZHAO W. X., LIU J., REN R. & WEN J.-R. (2024). Dense text retrieval based on pretrained language models : A survey. *ACM Transactions on Information Systems*, **42**(4), 1–60.

ZHAO X., HU X., SHAN Z., HUANG S., ZHOU Y., ZHANG X., SUN Z., LIU Z., LI D., WEI X. *et al.* (2025). Kalm-embedding-v2 : Superior training techniques and data inspire a versatile embedding model. *arXiv preprint arXiv :2506.20923*.

A Travaux Connexes - Suite

Extraction et recherche d'information à base de LLM. Les LLMs présentent de fortes capacités de compréhension du langage (Achiam *et al.*, 2023; Bai *et al.*, 2023; Dubey *et al.*, 2024; Yang *et al.*, 2025) et sont de plus en plus utilisés pour l'extraction d'information (ex. résumé automatique, reconnaissance d'entités nommées, analyse de sentiment) (Zhang *et al.*, 2024; Wang *et al.*, 2025, 2023). Dans notre contexte, (Kabongo & Guigue, 2025) propose d'extraire, à partir d'avis, des triplets énoncé–thème–sentiment pour évaluer la factualité des explications. En recherche d'information, les LLMs soutiennent à la fois le *dense retrieval* via embeddings sémantiques (Zhang *et al.*, 2025; Zhao *et al.*, 2024) et le reranking par cross-encoders (Sun *et al.*, 2023; Zhang *et al.*, 2025).

B Détails du Benchmark STAR

Nous fournissons ici une description détaillée du benchmark STAR (**Sta**tement **R**anking), comprenant le pipeline d'extraction et de vérification des énoncés, la procédure de clustering sémantique des paraphrases, et les statistiques des jeux de données construits.

B.1 Jeux de données STAR

	Toys	Clothes	Beauty	Sports
Statistiques Basiques				
Utilisateurs	18 594	39 371	22 361	35 594
Items	10 130	22 940	12 086	18 322
Interactions	155 992	274 635	198 225	294 513
Train	119 052	196 458	153 587	223 528
Eval	18 525	39 201	22 330	35 534
Test	18 415	38 976	22 308	35 451
Avant Clustering				
Énoncés Uniques	472 843	570 376	507 691	942 442
Positif	159 466	216 542	222 455	370 554
Négatif	98 229	146 886	112 446	193 553
Neutre	215 148	206 948	172 790	378 335
Min/Moy/Max par inter.	1/4.60/26	1/4.18/26	1/4.33/26	1/4.65/26
Min/Moy/Max par item	3/63.19/1101	5/46.35/1099	8/67.41/1649	8/71.08/3032
Min/Moy/Max par user	1/34.24/1701	1/27.79/536	2/37.45/1115	1/37.80/1758
Total triplets	718 313	1 149 483	858 733	1 371 531
Après Clustering				
Énoncés Uniques	281 664 (40.4%)	260 477 (54.3%)	229 448 (54.8%)	556 209 (41.0%)
Positif	80 476 (49.5%)	84 071 (61.2%)	81 774 (63.2%)	187 055 (49.5%)
Négatif	65 637 (33.2%)	75 635 (48.5%)	59 144 (47.4%)	134 118 (30.7%)
Neutre	135 551 (37.0%)	100 771 (51.3%)	88 530 (48.8%)	235 036 (37.9%)
Min/Moy/Max par inter.	1/4.57/26	1/4.15/26	1/4.28/26	1/4.62/25
Min/Moy/Max par item	3/56.81/877	5/42.49/770	7/58.38/1149	7/64.67/2081
Min/Moy/Max par user	1/32.51/1364	1/26.57/470	2/35.57/909	1/36.74/1526
Total triplets	712 963	1 142 256	850 332	1 361 389

TABLE 6 – Statistiques des jeux de données du benchmark STAR. Nous indiquons entre parenthèses le taux de réduction en % après clustering.

B.2 Extraction et vérification des énoncés

Extraction de candidats. Étant donné un texte d'avis t_{ui} écrit par l'utilisateur u pour l'item i, un LLM extrait un ensemble de candidats $\hat{\mathcal{S}}_{ui} = \{\hat{s}_1, \ldots, \hat{s}_{m_{ui}}\}$, chacun accompagné d'une étiquette de polarité. La polarité est préservée explicitement, car elle est centrale pour la recommandation explicable. Nous utilisons `Qwen3-14B` (Yang *et al.*, 2025) pour d'extraction, et reportons le prompt final employé pour cette étape dans la Table 7.

Vérification. Malgré un prompting soigné, une fraction des candidats reste non explicative, non atomique ou redondante. Une étape de vérification filtre ces sorties : à partir de $\hat{\mathcal{S}}_{ui}$, un second LLM produit l'ensemble final $\mathcal{S}_{ui} = \{s_1, \ldots, s_{n_{ui}}\}$. Nous employons `Qwen3-8B` (Yang *et al.*, 2025) pour la vérification, et fournissons le prompt final utilisé pour cette étape dans la Table 8.

B.3 Clustering des paraphrases

Notre approche de clustering sémantique procède en trois étapes :

Recherche de voisins approchés (ANN). Pour chaque énoncé $s \in \mathcal{S}^0$, nous calculons une représentation dense $\mathbf{s} = \Phi(s) \in \mathbb{R}^d$ à l'aide d'un encodeur Φ, puis effectuons une recherche ANN

dans l'espace d'embedding. La recherche est restreinte aux énoncés de même polarité p_s (pour éviter de rapprocher des sentiments opposés) et renvoie les K voisins les plus proches en similarité cosinus, notés $\mathrm{ANN}(s; K)$. Nous utilisons `Qwen3-Embedding-0.6B` (Zhang *et al.*, 2025)comme encodeur, avec normalisation ℓ_2. Les embeddings sont indexés avec Faiss (Douze *et al.*, 2025). Nous utilisons un index par polarité) et fixons $K = 128$.

Filtrage par paires. La proximité en embedding ne garantit pas l'équivalence paraphrastique. Nous filtrons donc les voisins en deux temps. D'abord, nous ne conservons que les paires dont la similarité cosinus dépasse τ_{pair} : $\mathcal{V}^0 = \bigcup_{s \in \mathcal{S}^0} \{\{s, s'\} : \ s' \in \mathrm{ANN}(s; K), \ \cos(\mathbf{s}, \mathbf{s}') \geq \tau_{\text{pair}}\}$. Ensuite, chaque paire candidate est réévaluée par un cross-encoder binaire Ψ : seules les paires dont la probabilité prédite dépasse 0.9 sont retenues comme paraphrases validées, formant l'ensemble $\mathcal{V}$. Nous fixons $\tau_{\text{pair}} = 0.9$ pour ne retenir que les paires très similaires, limitant le coût de la réévaluation. Nous utilisons `Qwen3-Reranker-0.6B` (Zhang *et al.*, 2025) comme cross-encoder, en privilégiant une haute précision.

Raffinement. À partir des paires validées $\mathcal{V}$, nous construisons un graphe de similarité non orienté $G = (\mathcal{S}^0, \mathcal{V})$. Les composantes connexes fournissent des clusters initiaux, mais peuvent contenir du bruit car de nombreuses paires intra-composante n'ont pas été explicitement validées. Pour chaque composante $G_\ell = (\mathcal{S}_\ell, \mathcal{V}_\ell)$, nous mesurons la cohésion par la similarité cosinus minimale intra-composante. Si elle dépasse τ_{intra}, la composante est retenue telle quelle. Sinon, elle est raffinée par l'Algorithme 1, qui la partitionne autour de *pivots* de haut degré et fusionne les clusters de pivots consécutifs dont la similarité dépasse τ_{remerge}. Dans chaque cluster final, le représentant retenu est l'énoncé le plus central (maximisant la similarité moyenne aux autres membres). L'ensemble canonique $\mathcal{S}$ est constitué de ces représentants. Nous fixons $\tau_{\text{intra}} = 0.85$ et $\tau_{\text{remerge}} = 0.90$ pour maintenir une forte cohésion intra-cluster.

Algorithm 1: Refinement of Non-Cohesive Component

Input: $G_\ell = (\mathcal{S}_\ell, \mathcal{V}_\ell)$, embeddings $\{\mathbf{s}\}_{s \in \mathcal{S}_\ell}$, threshold τ_{remerge}
Output: Refined clusters $\mathcal{C}_\ell$
Notations : $\mathcal{N}_\ell(s) = \{s' \in \mathcal{S}_\ell : (s, s') \in \mathcal{V}_\ell\}$;
$\mathcal{C}_\ell \leftarrow \emptyset$, $\mathcal{R} \leftarrow \mathcal{S}_\ell$, $\mathcal{A} \leftarrow \emptyset$, $\mathcal{P} \leftarrow \emptyset$;
while $\mathcal{R} \neq \emptyset$ **do**
  $p \leftarrow \arg\max_{s \in \mathcal{R}} |\mathcal{N}_\ell(s) \cap \mathcal{R}|$, $\mathcal{B} \leftarrow \{p\} \cup (\mathcal{N}_\ell(p) \cap \mathcal{R})$;
  if $\mathcal{A} = \emptyset$ **then**
    $\mathcal{A} \leftarrow \mathcal{B}$, $\mathcal{P} \leftarrow \{p\}$;
  else if $\min_{p' \in \mathcal{P}} \cos(\mathbf{p}, \mathbf{p}') \geq \tau_{\text{remerge}}$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{B}$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{p\}$;
  else
    $\mathcal{C}_\ell \leftarrow \mathcal{C}_\ell \cup \{\mathcal{A}\}$, $\mathcal{A} \leftarrow \mathcal{B}$, $\mathcal{P} \leftarrow \{p\}$;
  $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{B}$;
$\mathcal{C}_\ell \leftarrow \mathcal{C}_\ell \cup \{\mathcal{A}\}$;
return $\mathcal{C}_\ell$;

C Discussion

Classement d'énoncés vs. génération d'explications. Remplacer la génération libre par le classement d'énoncés répond à trois problèmes fondamentaux. Premièrement, la génération produit souvent des explications hallucinées ou génériques, tandis que le classement restreint les explications à des énoncés ancrés dans les avis, éliminant les hallucinations par construction. Deuxièmement, évaluer du texte généré est difficile à standardiser et reproduire entre métriques lexicales, sémantiques, apprises et LLM-as-a-judge, alors que le classement s'appuie sur des métriques de recherche d'information bien établies. Troisièmement, les explications sous forme de paragraphes mêlent plusieurs facteurs sans exposer leur contribution respective, tandis que le classement décompose les explications en unités atomiques et utilise les scores de pertinence pour modéliser l'importance de chaque facteur, permettant une analyse fine de ce qui justifie une recommandation et dans quelle mesure.

STAR pour le classement d'énoncés. STAR rend le classement d'énoncés concrètement significatif en garantissant que l'ensemble des énoncés candidats est *explicatif*, *atomique* et *unique*. Nous imposons l'explicativité et l'atomicité via un pipeline LLM d'extraction et de vérification, qui améliore substantiellement les deux propriétés, et nous imposons l'unicité par un clustering sémantique à grande échelle produisant des clusters cohésifs et bien séparés. Ces étapes combinées donnent un benchmark où les métriques de classement standard reflètent fidèlement la qualité des explications et mesurent si les modèles identifient les justifications les plus pertinentes pour chaque interaction utilisateur–item.

Comparaison global-level et item-level. Nos résultats révèlent une forte asymétrie entre les deux cadres d'évaluation. En global-level, EXPGCN obtient les meilleures performances, mais les baselines de fréquence item et globale (ITEMPOP, GLOBALPOP) restent compétitives, tandis que le signal historique utilisateur (USERPOP) s'effondre, suggérant un régime de récupération à grande échelle où les priors item dominent. En item-level, la dynamique s'inverse : USERPOP devient la baseline la plus forte en moyenne et peut égaler ou surpasser EXPGCN, tandis qu'ITEMPOP chute drastiquement, indiquant un régime principalement guidé par la personnalisation où la pertinence item est largement satisfaite par construction et où le défi est de prioriser les facteurs propres à l'utilisateur. Les tendances des hyperparamètres confortent cette interprétation : pour BPER+, les performances en global-level sont optimales pour un équilibre intermédiaire entre signaux utilisateur et item, tandis qu'en item-level elles s'améliorent à mesure que le poids utilisateur augmente. Au total, ces différences exposent un déficit de personnalisation clair : les gains en global-level ne se traduisent pas nécessairement en meilleur classement item-level, et les modèles actuels peinent à exploiter les signaux spécifiques à l'interaction au-delà de priors utilisateur grossiers.

Limites et perspectives. Bien que STAR impose des propriétés clés sur les énoncés, il repose sur un pipeline automatique d'extraction et de clustering, de sorte que des erreurs résiduelles et une consolidation imparfaite des paraphrases peuvent introduire du bruit dans la supervision et l'évaluation, et le benchmark peut hériter de biais présents dans les données d'avis. Au-delà de la pertinence binaire, étendre STAR avec une pertinence graduée (e.g., via la proéminence dans l'avis ou des annotations humaines) permettrait une évaluation plus nuancée, et intégrer des objectifs de diversité et de couverture pourrait mieux refléter la nature multi-facettes des explications. Enfin, la forte performance en item-level des signaux d'historique utilisateur simples met en évidence un déficit de personnalisation qui appelle des modèles apprenant explicitement des préférences utilisateur fines sur les facteurs explicatifs.

Extract atomic, factual, product-focused statements from a single product review.

INPUT
A single review text.

OUTPUT (STRICT)
Return ONLY valid JSON object with exactly one key :
`{"statements": [{"statement": "...", "sentiment": "..."}] }`

STATEMENT REQUIREMENTS
- No multi-proposition coordination : do not join two facts with “and/but/or”.
- Atomic : one fact per statement. If one statement is implied by another, remove the implied one.
- Present tense : rewrite into present tense.
- No personal information : statements must be general.
- Generalized : only product-related properties, behaviors, or outcomes affecting user experience.
- Product-focused : the subject must clearly refer to the product.
- If product name is unknown, use “the product . . . ”.
- Faithful : do not add information not in the review ; keep statements short.

SENTIMENT
- `"positive"` : desirable/appreciated property or outcome.
- `"negative"` : problem, failure, or disliked property.
- `"neutral"` : descriptive or mixed, without clear judgment.

Return ONLY the JSON object, with no preamble or surrounding text.

TABLE 7 – Prompt pour l’extraction d’énoncés candidats.

Clean a list of extracted review statements.

INPUT
`{"statements": [{"statement":"...","sentiment":...", ...]}}`

OUTPUT (STRICT)
Return ONLY : `{"statements": [{"statement":"...","sentiment":"...", ...]}}`

RULES
- Keep ONLY product-focused facts. Every output statement starts with “the product . . . ”.
- Present tense. Short. No guessing. No new info.
- Remove personal/story content : no “I/my/user/we/people/child/friends” no need to know if user/child or someone likes or hates the product.
- Drop anything about : buying intention, recommendations, social reactions, complaints/returns/support, seller/brand/store, shipping/delivery, nostalgia, or “I like/love/hate” without a concrete product attribute.
- Drop comparisons : any “better/worse/than”, “as good as”, “compared to”, or mentioning other products/-brands/models.
- Remove only intensifiers (very/super/really/extremely) while keeping the attribute.
- Remove redundancy : if one statement implies another, keep only the most specific. (No need to repeat similar facts).
- Sentiment must match the final statement.

Process the input JSON and return ONLY the cleaned JSON.

TABLE 8 – Prompt pour l’étape de vérification d’énoncés extraits.