# Let Me Explain - Knowledge-based Retrieval Augmented Generation for Agricultural Recommendation Explanations

Daan L. Di Scala[1,2][0000−0003−1548−6675] and Maaike H.T. de Boer[1][1111−2222−3333−4444]

[1] TNO Netherlands Organisation for Applied Scientific Research, Department Data Science, Kampweg 55, 3769 ZG Soesterberg, The Netherlands;
[2] Utrecht University, 3584 CS Utrecht, The Netherlands
{daan.discala,maaike.deboer}@tno.nl

**Abstract.** Recommender systems play an important role in providing items to users that align with their goals. However, explaining the relevance of recommended items remains challenging; a particularly important task for high-stakes domains like agriculture where both suitability and interpretability of recommendations is crucial. Knowledge Graph (KG)-based approaches can provide suitable recommendations, but lack fluent explanations. Recent Large Language Model (LLM)-based recommender systems excel in natural explanations, yet are limited in factuality or coherence. Combining strengths of LLM-based recommendations with knowledge presents a promising direction. We introduce KRAGGER (Knowledge Retrieval-Augmented Generation for Graph-based Explainable Recommendation), a composite framework combining KG- and LLM-based methods. We apply KRAGGER to an agricultural use case by testing its ability to recommend technologies to farmers while explaining provided matches. The contributions of this paper are: 1) KRAGGER's three recommendation methods: Graph-Graph, Graph-Attribute and Attribute-Attribute matching, and 2) KRAGGER's three explanation methods: Rule-Based (RB), LLM and RB-LLM, and 3) an agricultural recommendation KG dataset describing farmers and farm technologies. Recommendation performance of KRAGGER is measured quantitatively with catalog coverage and intra-list diversity on five different embedding models. Additionally, a user study is performed with domain experts to validate recommendation performance and measure simplicity, fluency, coherence and trustworthiness of explanations. Results show the fine-grained attribute matching recommendation performs best, which forms the basis of the highest rated RB-LLM rephrasing explanation method. Thus, KRAGGER shows promising capabilities for grounded, yet natural recommendation explanations, underscoring the importance of composite AI (KGs and LLMs) to support difficult decision making.

**Keywords:** Recommender Systems · Knowledge Graphs · Large Language Models · Retrieval-Augmented Generation · AI Explanations

## 1 Introduction

AI-driven recommendations are increasingly used; not only in our daily life to find hotels, restaurants or movies, but also in high-stakes domains like healthcare, judicial cases and agriculture [10,64]. Recommender systems, a type of advice-giving information retrieval system, support users in choosing among options that match their preferences or goals [5]. When users are faced with difficult decisions in sensitive contexts, providing high-quality advice is essential: recommendations should be both accurate and well-interpretable to avoid harmful outcomes [2,56,58]. Thus, creating trustworthy advice-providing AI requires attention on both performance and the quality of their explanations [21,36]. Simply put: in these domains, a recommendation without proper explanation is of little use, and vice versa: an explanation may be fluent and convincing, but if it is based on an unsuitable recommended item, it is not useful either.

One well-known approach for recommendation is Content-Based (CBR), where recommendations are based on matching the content of users and/or items [1]. This approach is particularly useful in 'cold start' situations, where little prior information is available [29], which is often the case in real-life high-stakes domains. Knowledge Graph (KG)-based approaches can provide suitable CBR, but often lack the capability of fluently explaining why particular matches are suggested [30]. Recently, the use of Large Language Models (LLMs) for recommendations has grown [32,53]. Approaches such as Retrieval-Augmented Generation (RAG) leverage the semantic strength of LLMs to both recommend content from sources and explain the match in a natural way. However, LLMs have limitations on coherence and factuality of their explanations [45].

To leverage the combined strengths of LLM-based CBR with KG-based CBR, we introduce KRAGGER (Knowledge Retriever Augmented Generation for Graph-based Explainable Recommendation), a recommendation framework that aims to provide knowledge-based RAG recommendation, enriched with recommendation explanations that utilizes rule-based templates and LLMs. This framework allows, instead of matching full free-text descriptions, users

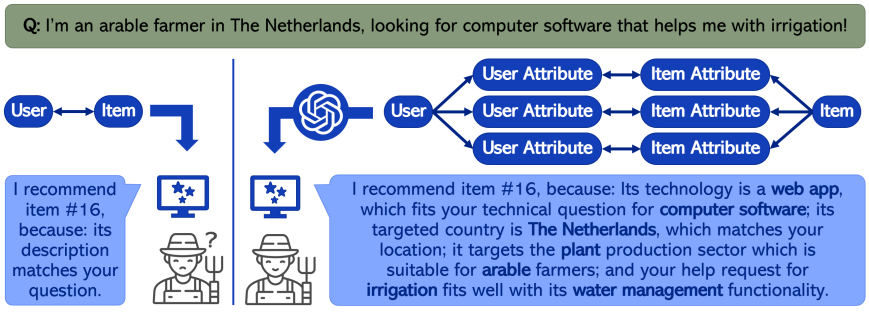



Fig. 1: An example of one-to-one user-item matching recommendation (left) versus attribute-based, LLM-processed recommendation with explanation (right).

and items to be described with more fine-grained information, aligned with structured knowledge representations in a KG. In this way, knowledge graphs can be integrated with RAG into composite CBR to provide advice that reflects both the user and item attributes. Figure 1 illustrates an example scenario in a high-stakes domain. On the left, a farmer is matched to an item in a one-to-one manner (user description to item description), which produces a limited recommendation and leaves the farmer with little explanatory insight. On the right, however, the farmer describes themselves in terms of attributes, which are aligned with item attributes and then processed by an LLM to provide a natural, more fine-grained explanation, leading to more insight. Throughout this paper, we use this case as a running example. This motivates two research questions:

**RQ1.** Does composite KG-LLM CBR lead to better recommendations?
**RQ2.** Does composite KG-LLM CBR lead to better explanations?

We compare a combination of knowledge graph- and LLM-based approaches to understand the impact on KRAGGER's performance. We hypothesize that the best performance is with more fine-grained attribute methods for both recommendation and explanation, and that combinations can be promising for trustworthy AI advice. We apply KRAGGER to an agricultural use case, as it is a domain with particularly pressing needs for explanations, as recommendations directly affect crop yields, resource use, and livelihoods. In such high-stakes contexts, explanations are critical not only for trust and accountability but also for enabling farmers to make informed decisions with significant financial, sustainable or social consequences [3,40]. To evaluate KRAGGER, we introduce an agricultural recommendation dataset. All data, code and results are made available in an open source repository for reproducibility purposes[3].

In the next section, we provide an overview of related knowledge- and LLM-based recommendation and evaluation works. In Section 3, we discuss KRAGGER and the experimental setup. In Section 4, we present our findings and discuss their implications and limitations in Section 5. Concluding, in Section 6 we summarize our research and provide suggestions for future work.

## 2 Related Work

Recommender systems can be categorized into Collaborative Filtering (CF), Content-based (CB), and hybrid approaches [1,20]. CF leverages user-item interactions, CB relies on item content, and hybrid techniques combine both. However, CF and hybrid approaches often struggle with the cold start problem, where little user or item information is available [15,29]. Information on users and items can be represented in knowledge graphs (KGs), which lead to KG-based recommender systems (KGBR). KGBR can be divided into three main

[3] https://github.com/daan-ds/KRAGGER

strategies: embedding-, connection-, or propagation-based. Embedding methods capture semantic relations but may miss higher-order interactions and are hard to interpret. Connection methods exploit graph connectivity for similarity but can lose ontological information. Propagation methods include entity relations, fully utilizing the graph's information [18,62].

With the rise of Large Language Models (LLMs), there has been a shift in recommender systems [53]. LLMs are used for improved representation (such as embeddings), improved understanding through pre- and post-recommendation explanations, and in a new paradigm of generative LLM-based recommendations. LLMs are used on their own [9,32,60], used to enhance existing recommender systems [33,48], used in a Retrieval-Augmented Generation (RAG) setting [6,8,61] or integrated in Graph RAG approaches [28,41,43]. K-RagRec [54] retrieves and re-ranks multi-hop subgraphs to augment LLM recommendations; RALLRec and RALLRec+ utilize LLM representation learning and reasoning for recommendation [31,57]; ER$^2$ALM provides a framework for attribute-aware RAG [55]; XRec [33] uses a collaborative tokenizer, information adapter and a hybrid method to integrate the collaborative filtering directly into the LLM to generate explanations. Based on these works, with KRAGGER we adopt attribute-based RAG CBR to deal with the cold start problem, describe users and items by KG attributes, and utilize LLMs as retriever and explainer components.

**Recommender System Evaluation.** Recommendation performance can be calculated with metrics like Accuracy, Recall, Hit-Rate or GDCN when a ground truth is available [63], which is why large-scale movie, book, or product recommendation benchmarks are widely-used [37,49]. However, when encountering the cold start problem, metrics that measure performance on more than accuracy are needed, such as Catalog Coverage (CC) or Intra-List Diversity (ILD) [13,24]. Furthermore, studies have been performed where users rated recommender systems on their performance [17,22,64]. To cover both aspects, we opt for a quantitative evaluation based on CC and ILD, as well as a user study with an expert group to validate KRAGGER's retrieval performance.

Besides evaluating recommenders on their recommendation performance, it is important to understand why certain items are recommended to a user. This is the field of explanations; an assignment of causal responsibility [19]. Different approaches towards explainability include data explainability, model explainability, feature-based techniques and example-based techniques [11]. Evaluating AI explanations is a challenging task [21,36,50]. We focus on these explanations from a system designer's perspective, defined as 'explanation goodness' [14,21], as we evaluate on textual model explainability to explain the relevancy of the recommendation. Simplicity measures how simple the explanation is, in terms of length and word complexity, as shorter, simpler explanations are often deemed more understandable [36]; Fluency measures how fluent or "natural" the explanation is [27,52]; Coherence measures how much the explanation is entailed by the input, or whether the explanation and recommendation contradict [36]. These metrics are suitable to assess different qualities of AI explanations and will therefore be used to evaluate the generation component of KRAGGER.

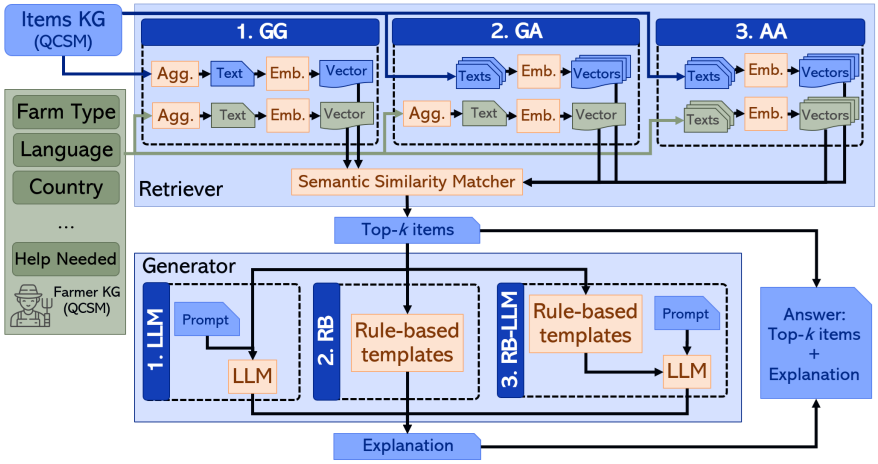



Fig. 2: KRAGGER architecture with retriever and generator components.

## 3 Method: KRAGGER

### 3.1 Recommendation - KRAGGER's Retriever Approaches

We compare three increasingly fine-grained retriever approaches that recommend items to users (Figure 2). **1) Graph-Graph (GG)**. This approach embeds the full descriptions of both the farmer subgraph and item subgraph and calculates the similarity for a relevancy score. For this approach, the texts of each node in the graph are first concatenated to achieve a description, before being embedded into one vector. It returns a list of items, sorted on semantic similarity $s(u, i)$ between user $u$ and item $i$. This approach acts as baseline, as it employs the typical retriever approach in RAG approaches of matching full chunks of texts. **2) Graph-Attribute (GA)**. This approach embeds each attribute of each item into separate vectors. Then, the similarity is calculated between each node vector and the graph vector of the aggregated user subgraph. All similarity scores are totaled for a similarity score, which leads to a sorted list of items. **3) Attribute-Attribute (AA)**. In the third approach, we embed both the attributes from the items and the farmer graph into into a shared vector space.

We restrict the semantic similarity matching to predefined attribute correspondences $\mathcal{M} \subseteq U \times I$ between user attributes $U$ and item attributes $I$, instead of taking the full Cartesian product of all attributes. This avoids a combinatorial explosion of pairings, removes semantically meaningless comparisons, and yields interpretable similarity scores. These matches determined based on overall semantic closeness of categorical attributes. For each pair $(u, i)$, the attribute-level score is defined as $S(u, i) = \max_{a_u \in A_u,\, a_i \in A_i} s(a_u, a_i)$, for each possible attribute per node. The overall similarity between user and item is $\mathrm{Sim}(u, i) = \frac{1}{|\mathcal{M}|} \sum_{(u,i)\in\mathcal{M}} S(u, i)$, with a weighted variant that emphasizes strong matches, with $S(u, i)^2$ instead of $S(u, i)$. This amplifies strong similarities and

Table 1: Explanation methods with corresponding texts (either LLM prompt or Rule-Based template text). Base Prompt: "A recommender system recommends an item to a user, based on provided information about the user. It recommends **{item}** to **{user}**."

| Expl. | Rec. | Text |
|---|---|---|
| **LLM** | | **{Base Prompt}**. Why would the item be recommended? Provide a clear explanation about why this item is recommended to this farmer. Start with "Because..." |
| **RB** | **GG** | Because the overall description of the user is similar to the overall description of the item. |
| | **GA** | Because the overall description of the user is similar to the attributes **{Item Attribute 1}**, ... , **{Item Attribute *n*}** of the item. |
| | **AA** | Because the attributes of the farmer description and the attributes of the items are similar: **{User Attribute 1}** is similar to **{Item Attribute 1}**, . . ., and **{User Attribute *n*}** is similar to **{Item Attribute *n*}** |
| **RB-LLM** | | **{Base Prompt}**. The item is recommended because: **{RB-GG / RB-GA / RB-AA}**. Rephrase the explanation to be a clear explanation about why this item is recommended to this user. Provide only the rephrased explanation. Start with "Because..." |

reduces weak ones. Similar to the other approaches, this leads to a sorted list of items, of which the top-$k$ items are presented as recommendations. For all three approaches, the cosine similarity is chosen to calculate semantic similarity, as is common practice [35].

### 3.2 Explanation - KRAGGER's Generation Methods

To explain why an item is recommended, we employ three explanation methods, as shown in the bottom half of Figure 2: **1) LLM.** This approach explains the top-$k$ matches using an LLM as text generator. This serves as a baseline, as it acts as a generator component in a default RAG system; **2) RB.** This approach outputs an explanation based on predefined rule-based texts describing the workings of each of the recommender systems. **3) RB-LLM**. This approach rephrases the output of the RB explanation using an LLM. For the corresponding LLM prompts and rule-based templates, see Table 1. For methods 1 and 3, we utilize one of the most state-of-the-art LLMs at the time, which is OpenAI's GPT-4.1 [38], and set its temperature to 0.3 to mitigate hallucinations.

### 3.3 Experimental Setup

**Item Data (Ontology and Knowledge Graph)** The agricultural items to be recommended are from a collection called the QuantiFarm dataset [44], which stems from three different real-world data sources from various collections of farm technologies [40], extracted from files with different file types (Excel, JSON), structures, and naming conventions. To unify these datasets , the QuantiFarm Common Semantic Model (QCSM) is created. The QCSM includes concepts and relations regarding farmers, farms and parcels, as well as attributes of the items to be recommended.

Table 2: Embedding models with MTEB ranking and output dimensions

| Name | Exact Model | Ranking | Dimensions |
|---|---|---|---|
| **QWEN** | Qwen/Qwen3-Embedding-4B [59] | 4 | 2560 |
| **E5** | intfloat/e5-mistral-7b-instruct [51] | 15 | 4096 |
| **GPT** | text-embedding-3-small [39] | 48 | 1536 |
| **MINI** | transformers/all-MiniLM-L6-v2 [23] | 107 | 384 |
| **AGRI** | recobo/agri-sentence-transformer [46] | - | 512 |

In total, the ontology consists of 31 classes and 185 properties. The QCSM allows to transform farm data according to the RDF standard and imports from other ontologies, such as the agricultural Ploutos [4] and SAREF4AGRI [7,47] ontologies, as well as GS1 for country codes [16]. The ontology is created according to available ontology engineering methodologies (SAMOD [42], SABiO [12], TDD [25]) and based on data requirements to determine the scope of the ontology, and which existing ontologies could be reused. Each item's subgraph describes both general information (e.g., name, title, description, keywords) as well as categorical information (e.g., benefits, functionalities, targeted sector). The full knowledge graph consists of 16.216 triples describing a total of 532 items. The ontology and knowledge graph are available in the repository[3].

**User Data.** In this use case, users can be described as farmers with six categorical attributes, $U = \{F, L, C, T, AD, AR\}$, where $F$ = Farm Type, $La$ = Language, $C$ = Country, $T$ = Technology Competence, $AD$ = Achievement Desired, $AR$ = Assistance Required. From these six possible user attributes, we generate a dataset consisting of 1000 **unsure users** $U_u$ and 1000 **targeted users** $U_t$, to simulate two different types of farmers that do or do not yet know what sort of items they are interested in. Unsure users $u_u \in U_u$ are defined $u_u = \langle f, t, l \rangle$, with $f \in F$, $t \in T$, $l \in L \cup C$. Targeted users $u_t \in U_t$ are defined $u_t = \langle f, t, l, a \rangle$. Here, $f \in F,\ t \in T,\ l \in L \cup C,\ a \in AD \cup AR$. All users are assigned one of each possible attributes, randomly sampled from the existing options in the knowledge graph. We specifically sample from the unions $L \cup C$ and $AD \cup AR$ for consistency reasons, to avoid conflicting information that would be unlikely to appear in actual user scenarios (e.g., an Icelandic farmer who speaks only Japanese). Following the example from Figure 1, an unsure user would be an arable farmer ($F$) from the Netherlands ($C$), who can work with software ($T$). A targeted user would also mention they are looking for help with crop irrigation ($AR$), or that they want to improve their environmental sustainability ($AD$). Based on these items and users, for the AA method the set of attribute correspondences for users $u$ and items $i$ becomes $\mathcal{M} = \{(u_{\text{Farm Type}}, i_{\text{Targeted Sector}}),\ (u_{\text{Language}},\ i_{\text{Language}}),\ (u_{\text{Country}},\ i_{\text{Country}}),\ (u_{\text{Tech Competence}}, i_{\text{Digital Form}}),\ (u_{\text{Achievement Desired}}, i_{\text{Benefit}}),\ (u_{\text{Assistance Required}}, i_{\text{Functionality}})\}$.

**Models.** To evaluate the different recommendation approaches, we compare five embedding models, as shown in Table 2. We compare embedders that rank differently on various tasks in the Embedding Leaderboard (MTEB) [34].

We include two smaller embedders, among which AGRI to evaluate whether finetuning on agricultural data has any significant effect for our use case.

**Evaluation Metrics.**
*Quantitative.* We quantitatively evaluate the recommendation performance on two suitable metrics. We measure on different top-$k$ recommendations ($k$@{1, 3, 5, 10, 20}). For high-stakes advice, the first few items can be crucial, which makes $k$@{1, 3} particularly interesting for this study. Yet, $k$@{5, 10, 20} provide a solid overview of total performance on a broader set of recommended items.

**Catalog Coverage (CC@$k$)** measures how much of the total set of DATSs is covered [13], defined as: $\text{CC@}k = \left|\bigcup_{u \in U} \text{Rec}_u^k\right| / |C|$. Here, $\bigcup_{u \in U} \text{Rec}_u^k$ is the union of all unique top-$k$ items recommended across all users $u \in U$, and $|C|$ is the total number of items. This metric indicates how well the recommendations make use of its full catalog, where higher CC@$k$ indicates avoiding bias and ensuring sufficient items get exposure.

**Intra-List Diversity (ILD@$k$)** measures the diversity within the top-$k$ recommended items [26]. It is calculated by computing the average pairwise dissimilarity between the top-$k$ recommended items, which is then averaged across all users:

$\text{ILD@}k = \frac{1}{|U|} \sum_{u \in U} \left( \frac{1}{\binom{k}{2}} \sum_{\{i,j\} \subset \text{Rec}_u^k} d(i,j) \right)$. In this case, $d(i,j) = 1 - cos(i,j)$ is the dissimilarity between items $i$ and $j$, based on cosine similarity, $\binom{k}{2}$ is the number of unique item pairs in a list of length $k$ and $\{i,j\} \subset \text{Rec}_u^k$ are all unordered pairs of items without repetition. This metric shows how varied the recommended items are, where higher ILD@$k$ indicates a reduced redundancy by mitigating bias towards repetitive or similar content.
*User Study.* We validate the performance of the recommender systems through a user study with agricultural experts, consisting of a two-part survey, questioning 1) the performance of the retriever component - rating a sample of 30 different recommendations (sets of items at different $k$: #1, #3, #5, #10, #20 and #532, to check whether their ratings are in line with the matched rankings) for different users on a 7-points Likert scale (very unfitting - very fitting), and 2) the quality of the generation component's output - on the running example recommendation (Figure 1), using **Simplicity**, **Fluency** and **Coherence** (defined in Section 2) and perceived **Trustworthiness** on a 7-points Likert scale. To ensure quality of measurements, each participant is asked to self-rate their confidence on their knowledge and expertise within the agricultural domain. For the full survey, including instructions and questions, see repository[3].

## 4 Results

We run a total of 50.000 recommendation experiments (5 models × 5 methods (GG, GA, GA weighted, AA, AA weighted) × (1000 $U_u$ + 1000 $U_t$ users) on the knowledge graph of 532 items. We measure the quantitative metrics on the top-$k$@{1, 3, 5, 10, 20} recommended sets. A negligible difference is found between

weighted and unweighted variants, so only unweighted methods are presented. For the user study, the recommended items are uniformly sampled.

### 4.1 Quantitative Results

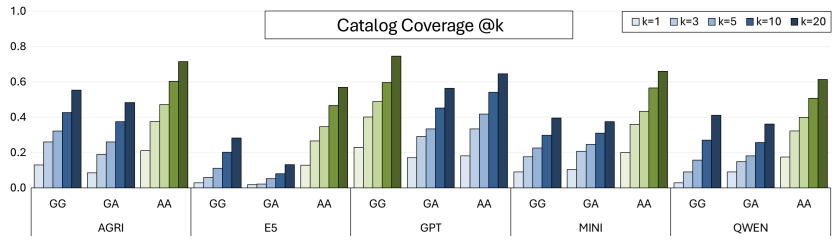



Fig. 3: Catalog Coverage results. Highest performing scores are marked green.

**Catalog Coverage (CC)**. Results for CC are shown in Figure 3. As expected, for all settings, a lower $k$ (fewer recommended items) leads to a lower coverage. For all models and all $k$, AA outperforms GG, except for GPT. GN scores lowest in all cases. For all settings, smaller models AGRI and MINI outperform larger models QWEN and E5. $\text{GG}_{\text{GPT}}$ scores highest, with $k@1 = 0.23$ and $k@20 = 0.746$. This is comparable to $\text{AA}_{\text{AGRI}}$'s scores of $k@1 = 0.21$ and $k@20 = 0.716$. AGRI slightly outperforms MINI, with a small difference ($<0.16$).

**Intra-List Diversity (ILD)**. Results for ILD are shown in Figure 3. $k = 1$ is left out as no diversity can be calculated. As expected, ILD scores increase overall with higher $k$. In all cases, GG achieves highest diversity, with $\text{GG}_{\text{QWEN}}@1{=}0.56$ as highest score, although differences between approaches are small ($<0.1$) for most models (AGRI, GPT, MINI). AGRI outperforms MINI slightly with GG, but with GA and AA the difference is negligible ($<0.02$). To understand the maximum possible ILD, an additional study was performed, yielding an average distribution of dissimilarity between all 532 items to peak at 0.6. Full details on the quantitative results and distribution study are found in the repository[3].

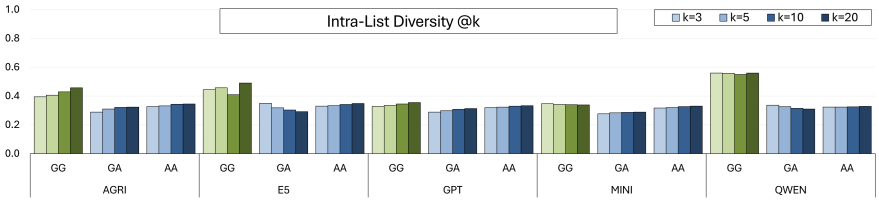



Fig. 4: Intra-List Diversity results. Highest performing scores are marked green.

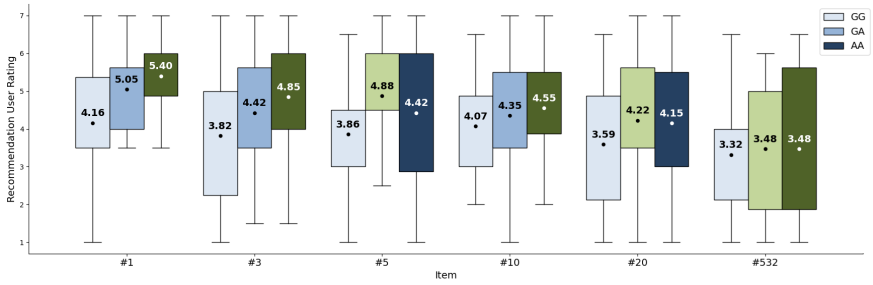


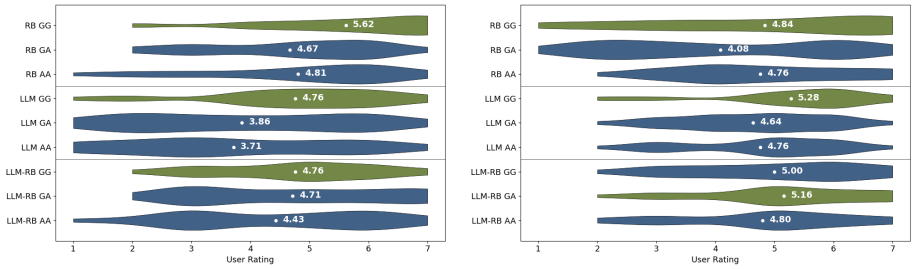

Fig. 5: User method ratings. Top scores are marked green.

### 4.2 User Study Results

A total of $n = 49$ voluntary experts within the agricultural domain participated in the user study. From this, 16 results were filtered out due to incomplete results. From the remaining 33, participants rated themselves an average of 5.4 out of 7 on their knowledge and expertise in the agricultural domain. Answers about their role within the domain show a diverse set of experts, a subset: an agricultural advisor and a pig monitoring researcher, people working in aquaculture and agricultural machinery industry, and multiple instances of agricultural policy advisors, agricultural technological providers and agronomy leads.

**Recommendation Results.** The expert ratings on an aggregation of 30 recommendations per expert (990 recommendations in total) are shown in Figure 5. The items that occurred as the number #1 recommended item were rated higher than those on lower positions, with the lowest recommended item scoring the least fitting ratings by users (GG scoring 3.32). As most recommenders scored $>$4 on the 7-point Likert scale on items #1-#20, most participants were positive about the provided recommendations. In all cases, the GG approach scores the lowest fitting score, and GA and AA alternate as top scoring approaches. On top $k$ items (#1, #3) however, GA outperforms the other methods, with AA reaching an average score of 5.4. A similar evaluation is performed with the data split over the five embedders, which shows MINI ($k$@1 = 5.46) and QWEN ($k$@1 = 5.25) to be the models with the best rated recommendations, yet little difference between model results is evident. Full results, including an additional figure of the model split, are found in the repository[3].

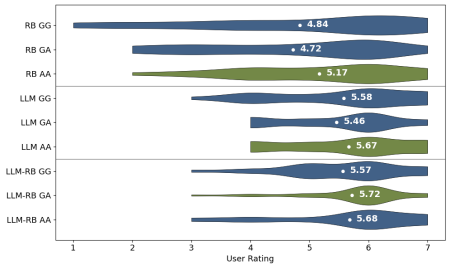
**Explanation Results.** All explanations were evaluated by the experts, as shown in Figures 6a-6d. The rule-based (RB) approach outperforms others in terms of simplicity ($\text{RB}_{\text{GG}} = 5.62$). The LLM and RB-LLM approaches score best on fluency ($\text{LLM}_{\text{GG}} = 5.28$), coherence ($\text{RB-LLM}_{\text{GA}} = 5.72$) and trustworthiness ($\text{LLM}_{\text{GA}} = 5.70$). For simplicity and fluency, the majority of best explanations are based on the graph-graph matching. For coherence and trustworthiness, all best explanations are based on graph-attribute and graph-graph matching. Averaging over all four metrics, RB-LLM performs best, specifically with attribute-based matching ($\text{RB-LLM}_{\text{GA-AVG}} = 5.22$, $\text{RB-LLM}_{\text{AA-AVG}} = 5.14$).



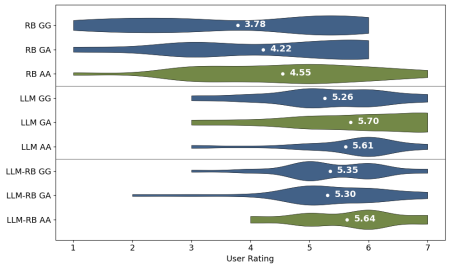

(a) Simplicity




(b) Fluency




(c) Coherence




(d) Trustworthiness

Fig. 6: User ratings on explanation metrics. Top scores are marked green.

## 5 Discussion

The quantitative results of the recommendation show that for coverage of recommended items, attribute-based approaches have a positive impact, whereas for diversity no large differences are found compared to the baseline approach. For recommendation performance, the type of embedding model used has little difference. Notably, the agricultural fine-tuned smaller embedder outperformed both larger and non-finetuned smaller models on quantitative results. This indicates that with the right fine-grained approach, smaller models that rank lower on MTEB (Table 2), do benefit from a fine-grained matching approach. Similarly, large (less cost-effective) models are not always needed to match attributes. These findings are supported by the user study, which showed for top $k$ items, attribute-based recommendations were rated as more fitting. and that the type of model had less of an impact. This leads us to answer **RQ1** (Section 1) that composite KG-LLM CBR can indeed lead to better recommendations.

The user study shows a trade-off of different qualities for explanations between the different approaches. LLM explanations are most often perceived as complex, and RB lacks fluency. The RB explanations being rated the most simple (Figure 6a) is in line with expectation, as they are on average shorter and more to the point than explanations generated with LLMs. Interestingly, the methods involving LLMs are top-rated on coherence, which is an important factor to base high-stakes decisions on. While RB is designed to coherently describe

the matching process (Table 1), it is not perceived this way. Figure 6d highlights remarkably many experts rating the rephrasing approach to be the most trustworthy, without any negative ratings. Fluency and coherence are mostly correlating with trustworthiness scores, indicating that these metrics are a better indicator for perceived trust than simplicity. Overall, the rephrasing composite RB-LLM explanation shows the most potential, adhering to the most metrics with high performance. This leads us to answer **RQ2** (Section 1) that composite KG-LLM CBR shows promise to lead to better explanations. Interestingly, the LLM and LLM-RB explainers provide adequate explanations, yet the LLM is not explicitly instructed to focus on each of the four explanation metrics (Table 1). It could be worthwhile to further investigate the effect on explanation goodness when specifically prompting with rules on what makes a good explanation.

We opted to select only agricultural professionals instead of external participants to ensure a high quality of perceived usefulness of the recommendations. Participants provided additional comments regarding their rating process, e.g., being more lenient towards English items by making the assumption that farmers with different background have some base level understanding of English. This raises an interesting topic on personal assumptions, both for validation with participants as well as for further designing tailored recommendations. KRAGGER only employs semantic similarity, so it is not able to keep these types of insightful nuances in mind. This would require additional rules or knowledge being incorporated into the recommendation process.

While this work currently targets agricultural use case, the KRAGGER framework could be flexibly applied to different domains, given that users describe themselves based on attributes that is not one-to-one the same as the item data (e.g., movie watchers describing their taste, mood, and context) while wanting to receive a textual explanation behind matches. Future work could expand on this framework with other recommendation benchmarks in other domains.

Limitations of this work include technical and environmental limitations, such as the decisions to use a small version of the GPT embedder, and keeping the dataset size at 2000 users and a sparse KG of 532 items. The latter affects recommendation performance, as not for all users a fitting item can be found. Furthermore, we focus on the LLM capabilities for recommendation and subsequent explanation-driven advice. To do so, we have chosen a baseline (GG-LLM) to be equivalent to standard Retrieval-Augmented Generation (RAG) approaches. However, other baselines, such as using different types of retriever or traditional recommendation models, could be explored for a more extensive comparison. Furthermore, we want to emphasize that LLMs ought to be used responsibly, for example for sustainability reasons. While the retrieval methods require a one-time only embedding step for the knowledge graph items, the best performing Attribute-Attribute method requires multiple embeddings to be produced for each user. Next to that, the attribute-based recommendation uses the fine-grained knowledge on each item that is stored in the graph. Extensions of KRAGGER should be explored that further exploit the structural properties of the knowledge graph, such as interconnectedness and multi-hop relations,

through other graph-based RAG approaches. Additionally, the explanations were evaluated on four metrics, but alternative or more metrics could have been chosen. Another limitation of the explainer methods is that they explain why an item is a good match based on positive information, but do not explicitly take any mismatching or negative information into account. They furthermore do not convey any explicit (un)certainty about their recommendations. Based on this, future work could look into further improving on the completeness and persuasiveness of the recommendations. Alternatively, a comparison of different LLM generators or prompts could be performed.

## 6 Conclusion and Future Work

We have proposed KRAGGER, a recommendation framework that combines knowledge graphs with retrieval-augmented generation (RAG) approaches, which consists of three recommender methods (Graph-Graph, Graph-Attribute, and Attribute-Attribute matching) and three explanation methods (Rule-Based (RB), LLM and LLM-RB rephrasing). We have measured KRAGGER's ability in an agricultural use case to recommend technologies to farmers while textually explaining the relevancy of provided matches based on the introduced agricultural knowledge graph. We have evaluated its performance based on quantitative metrics (catalog coverage, intra-list diversity), and performed a user study to assess both its recommendation performance and four qualitative metrics on explanation goodness (simplicity, fluency, coherence and trustworthiness).

Results show the fine-grained attribute-attribute matching approaches perform best on recommendation, which simultaneously forms the basis of the best performing LLM-RB rephrasing explanation method. Thus, KRAGGER shows promising capabilities for grounded, yet natural advice. This underscores the importance of leveraging the strengths of knowledge, rule-based approaches and language models, without overly relying on one or the other approach. Future work could expand on this towards other (high-stakes) domains, further extending on performance metrics with accuracy and honesty, or with more extensive user studies on the perceived effects on user trust. With this, we believe to have made meaningful steps towards trustworthy AI advice systems.

**Acknowledgments.** This research has been partially funded by the European Union project QuantiFarm: 'Assessing the impact of digital technology solutions in agriculture in real-life conditions', under the Grant ID 101059700. We would furthermore like to thank our colleagues Jack Verhoosel for his work on the QCSM ontology, David de Best for working on the QuantiFarm recommender tool, as well as Christopher Brewster and Caroline van der Weerdt for assisting with the data collection.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References


1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering **17**(6), 734–749 (2005)
2. Bohannon, M.: Lawyer Used ChatGPT In Court—And Cited Fake Cases. A Judge Is Considering Sanctions. https://www.forbes.com/sites/mollybohannon/2023/06/08/lawyer-used-chatgpt-in-court-and-cited-fake-cases-a-judge-is-considering-sanctions/ (2023), [Accessed 01-01-2026]
3. Brewster, C.: Planetary health and digital Agriculture: Rethinking technological innovation. In: Short Paper Proceedings of the 11th International Conference on Information and Communication Technologies in Agriculture, Food & Environment (HAICTA 2024) (2024)
4. Brewster, C., Kalatzis, N., Nouwt, B., Kruiger, H., Verhoosel, J.: Data sharing in agricultural supply chains: Using semantics to enable sustainable food systems. Semantic Web **15**(4), 1207–1237 (2024)
5. Burke, R., Felfernig, A., Göker, M.H.: Recommender systems: An overview. AI Magazine **32**(3), 13–18 (2011)
6. Contal, E., McGoldrick, G.: Ragsys: Item-cold-start recommender as rag system. arXiv preprint arXiv:2405.17587 (2024)
7. Daniele, L., den Hartog, F., Roes, J.: Created in close interaction with the industry: the smart appliances reference (SAREF) ontology. In: International Workshop Formal Ontologies Meet Industries. pp. 100–112. Springer (2015)
8. Di Palma, D.: Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In: Proceedings of the 17th ACM Conference on Recommender Systems. pp. 1369–1373 (2023)
9. Di Palma, D., Biancofiore, G.M., Anelli, V.W., Narducci, F., Di Noia, T., Di Sciascio, E.: Evaluating chatgpt as a recommender system: A rigorous approach. arXiv preprint arXiv:2309.03613 (2023)
10. Dunning, R.E., Fischhoff, B., Davis, A.L.: When do humans heed AI agents' advice? When should they? Human Factors **66**(7), 1914–1927 (2024)
11. Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., et al.: Explainable ai (xai): Core ideas, techniques, and solutions. ACM Computing Surveys **55**(9), 1–33 (2023)
12. Falbo, R.: Sabio: Systematic approach for building ontologies. CEUR Workshop Proceedings **1301** (2014)
13. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 257–260 (2010)
14. Glass, D.H.: How good is an explanation? Synthese **201**(2), 53 (2023)
15. Gope, J., Jain, S.K.: A survey on solving cold start problem in recommender systems. In: 2017 International Conference on Computing, Communication and Automation (ICCCA). pp. 133–138. IEEE (2017)
16. GS1 Web Vocabulary. https://ref.gs1.org/voc/, [Accessed 01-01-2026]
17. Gunawardana, A., Shani, G., Yogev, S.: Evaluating recommender systems. In: Recommender systems handbook, pp. 547–601. Springer (2012)
18. Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., He, Q.: A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering **34**(8), 3549–3568 (2020)

19. Halpern, J.Y., Pearl, J.: Causes and explanations: A structural-model approach. part ii: Explanations. The British journal for the philosophy of science (2005)
20. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: 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. pp. 639–648 (2020)
21. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Measures for explainable ai: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-ai performance. Frontiers in Computer Science **5**, 1096257 (2023)
22. Hu, R., Pu, P.: A study on user perception of personality-based recommender systems. In: International conference on user modeling, adaptation, and personalization. pp. 291–302. Springer (2010)
23. HuggingFace Page all-MiniLM-L6-v2 Embedding Model. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, [Accessed 01-01-2026]
24. Jadon, A., Patil, A.: A comprehensive survey of evaluation techniques for recommendation systems. In: International Conference on Computation of Artificial Intelligence & Machine Learning. pp. 281–304. Springer (2024)
25. Keet, C.M., Ławrynowicz, A.: Test-driven development of ontologies. In: European Semantic Web Conference. pp. 642–657. Springer (2016)
26. Kunaver, M., Požrl, T.: Diversity in recommender systems–A survey. Knowledge-based systems **123**, 154–162 (2017)
27. Lee, D., Whang, T., Lee, C., Lim, H.: Towards reliable and fluent large language models: Incorporating feedback learning loops in qa systems. arXiv preprint arXiv:2309.06384 (2023)
28. Li, Y., Zhang, X., Luo, L., Chang, H., Ren, Y., King, I., Li, J.: G-refer: Graph retrieval-augmented large language model for explainable recommendation. In: Proceedings of the ACM on Web Conference 2025. pp. 240–251 (2025)
29. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the cold start problem in recommender systems. Expert systems with applications **41**(4), 2065–2073 (2014)
30. Lully, V., Laublet, P., Stankovic, M., Radulovic, F.: Enhancing explanations in recommender systems with knowledge graphs. Procedia Computer Science **137**, 211–222 (2018)
31. Luo, S., Xu, J., Zhang, X., Wang, L., Liu, S., Hou, H., Song, L.: Rallrec+: Retrieval augmented large language model recommendation with reasoning. arXiv preprint arXiv:2503.20430 (2025)
32. Lyu, H., Jiang, S., Zeng, H., Xia, Y., Wang, Q., Zhang, S., Chen, R., Leung, C., Tang, J., Luo, J.: Llm-rec: Personalized recommendation via prompting large language models. arXiv preprint arXiv:2307.15780 (2023)
33. Ma, Q., Ren, X., Huang, C.: Xrec: Large language models for explainable recommendation. arXiv preprint arXiv:2406.02377 (2024)
34. Massive Text Embedding Benchmark Embedding Leaderboard. https://huggingface.co/spaces/mteb/leaderboard/ (2025), [Accessed 01-01-2026]
35. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
36. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence **267**, 1–38 (2019)
37. Ni, J., Li, J., McAuley, J.: 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). pp. 188–197 (2019)

38. OpenAI Platform GPT-4.1 Model. https://platform.openai.com/docs/models/gpt-4.1/, [Accessed 01-01-2026]
39. OpenAI Platform Vector Embedding Models. https://platform.openai.com/docs/guides/embeddings, [Accessed 01-01-2026]
40. Papadopoulos, G., Papantonatou, M.Z., Uyar, H., Kriezi, O., Mavrommatis, A., Psiroukis, V., Kasimati, A., Tsiplakou, E., Fountas, S.: Economic and environmental benefits of digital agricultural technological solutions in livestock farming: A review. Smart Agricultural Technology p. 100783 (2025)
41. Papageorgiou, G., Sarlis, V., Maragoudakis, M., Tjortjis, C.: Hybrid Multi-Agent GraphRAG for E-Government: Towards a Trustworthy AI Assistant. Applied Sciences **15**(11), 6315 (2025)
42. Peroni, S.: Samod: an agile methodology for the development of ontologies. In: Proceedings of the 13th OWL: Experiences and Directions Workshop and 5th OWL reasoner evaluation workshop (OWLED-ORE 2016). pp. 1–14 (2016)
43. Qiu, Z., Luo, L., Pan, S., Liew, A.W.C.: Unveiling user preferences: A knowledge graph and llm-driven approach for conversational recommendation. arXiv preprint arXiv:2411.14459 (2024)
44. QuantiFarm Homepage. https://quantifarm.eu/ (2025), [Accessed 01-01-2026]
45. Raczyński, J., Lango, M., Stefanowski, J.: The problem of coherence in natural language explanations of recommendations. arXiv preprint arXiv:2312.11356 (2023)
46. Recobo Agri Sentence Transformer Model. https://huggingface.co/recobo/agri-sentence-transformer, [Accessed 01-01-2026]
47. SmartM2M, E.: Extension to SAREF; Part 6: Smart Agriculture and Food Chain Domain. ETSI Technical Specification **103**, 410–6 (2024)
48. Tian, C., Hu, B., Gan, C., Chen, H., Zhang, Z., Yu, L., Liu, Z., Zhang, Z., Zhou, J., Chen, J.: Reland: Integrating large language models' insights into industrial recommenders via a controllable reasoning pool. In: Proceedings of the 18th ACM Conference on Recommender Systems. pp. 63–73 (2024)
49. Vente, T., Wegmeth, L., Said, A., Beel, J.: From clicks to carbon: the environmental toll of recommender systems. In: Proceedings of the 18th ACM Conference on Recommender Systems. pp. 580–590 (2024)
50. van der Waa, J., Nieuwburg, E., Cremers, A., Neerincx, M.: Evaluating xai: A comparison of rule-based and example-based explanations. Artificial intelligence **291**, 103404 (2021)
51. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F.: Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533 (2022)
52. Wang, L., Lyu, C., Ji, T., Zhang, Z., Yu, D., Shi, S., Tu, Z.: Document-level machine translation with large language models. arXiv preprint arXiv:2304.02210 (2023)
53. Wang, Q., Li, J., Wang, S., Xing, Q., Niu, R., Kong, H., Li, R., Long, G., Chang, Y., Zhang, C.: Towards next-generation LLM-based recommender systems: A survey and beyond. arXiv preprint arXiv:2410.19744 (2024)
54. Wang, S., Fan, W., Feng, Y., Ma, X., Wang, S., Yin, D.: Knowledge graph retrieval-augmented generation for LLM-based recommendation. arXiv preprint arXiv:2501.02226 (2025)
55. Wei, C., Duan, K., Zhuo, S., Wang, H., Huang, S., Liu, J.: Enhanced recommendation systems with retrieval-augmented large language model. Journal of Artificial Intelligence Research **82**, 1147–1173 (2025)
56. Wells, K.: An eating disorders chatbot offered dieting advice, raising fears about AI in health. https://www.npr.org/sections/health-

shots/2023/06/08/1180838096/an-eating-disorders-chatbot-offered-dieting-advice-raising-fears-about-ai-in-hea (2023), [Accessed 01-01-2026]
57. Xu, J., Luo, S., Chen, X., Huang, H., Hou, H., Song, L.: Rallrec: Improving retrieval augmented large language model recommendation with representation learning. In: Companion Proceedings of the ACM on Web Conference 2025. pp. 1436–1440 (2025)
58. Yagoda, M.: Airline held liable for its chatbot giving passenger bad advice - what this means for travellers. https://www.bbc.com/travel/article/20240222-air-canada-chatbot-misinformation-what-travellers-should-know (2024), [Accessed 01-01-2026]
59. Zhang, Y., Li, M., Long, D., Zhang, X., Lin, H., Yang, B., Xie, P., Yang, A., Liu, D., Lin, J., Huang, F., Zhou, J.: Qwen3 embedding: Advancing text embedding and reranking through foundation models. arXiv preprint arXiv:2506.05176 (2025)
60. Zhang, Y., Ding, H., Shui, Z., Ma, Y., Zou, J., Deoras, A., Wang, H.: Language models as recommender systems: Evaluations and limitations (2021)
61. Zhou, H., Gu, H., Liu, X., Zhou, K., Liang, M., Xiao, Y., Govindan, S., Chawla, P., Yang, J., Meng, X., et al.: The efficiency vs. accuracy trade-off: Optimizing rag-enhanced llm recommender systems using multi-head early exit. arXiv preprint arXiv:2501.02173 (2025)
62. Zhou, W., Han, W.: Personalized recommendation via user preference matching. Information Processing & Management **56**(3), 955–968 (2019)
63. Zhu, J., Dai, Q., Su, L., Ma, R., Liu, J., Cai, G., Xiao, X., Zhang, R.: Bars: Towards open benchmarking for recommender systems. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2912–2923 (2022)
64. Ziegfeld, L., Di Scala, D., Cremers, A.H.: The effect of preference elicitation methods on the user experience in conversational recommender systems. Computer Speech & Language **89**, 101696 (2025)