Generating Usage-related Questions for Preference Elicitation in Conversational Recommender Systems

IVICA KOSTRIC and KRISZTIAN BALOG, University of Stavanger, Norway
FILIP RADLINSKI, Google, UK

A key distinguishing feature of conversational recommender systems over traditional recommender systems is theirability to elicit user preferences using natural language. Currently, the predominant approach to preference elicitation is to ask questions directly about items or item attributes. Users searching for recommendations may not have deep knowledge of the available options in a given domain. As such, they might not be aware of key attributes or desirable values for them. However, in many settings, talking about the *planned use* of items does not present any difficulties, even for those that are new to a domain. In this article, we propose a novel approach to preference elicitation by asking implicit questions based on item usage. As one of the main contributions of this work, we develop a multi-stage data annotation protocol using crowdsourcing, to create a high-quality labeled training dataset. Another main contribution is the development of four models for the question generation task: two template-based baseline models and two neural text-to-text models. The template-based models use heuristically extracted common patterns found in the training data, while the neural models use the training data to learn to generate questions automatically. Using common metrics from machine translation for automatic evaluation, we show that our approaches are effective in generating elicitation questions, even with limited training data. We further employ human evaluation for comparing the generated questions using both pointwise and pairwise evaluation designs. We find that the human evaluation results are consistent with the automatic ones, allowing us to draw conclusions about the quality of the generated questions with certainty. Finally, we provide a detailed analysis of cases where the models show their limitations.

CCS Concepts: • **Information systems** → **Recommender systems**; *Users and interactive retrieval;*

Additional Key Words and Phrases: Conversational recommender systems, preference elicitation, question generation

ACM Reference format:
Ivica Kostric, Krisztian Balog, and Filip Radlinski. 2024. Generating Usage-related Questions for Preference Elicitation in Conversational Recommender Systems. *ACM Trans. Recomm. Syst.* 2, 2, Article 12 (April 2024), 24 pages.
https://doi.org/10.1145/3629981

1 INTRODUCTION

Traditionally, recommender systems predict users' preference towards an item by performing offline analysis of past interaction data (e.g., click history, past visits, item ratings) [14]. These

An early version of this work has been published as a late-breaking results article at RecSys'21 [24].

Authors' addresses: I. Kostric and K. Balog, University of Stavanger, Postboks 8600, 4036 Stavanger; e-mail: ivica.kostric@uis.no, krisztian.balog@uis.no; F. Radlinski, Google, 6 Pancras Square, London, N1C 4AG; e-mail: filiprad@google.com.

2770-6699/2024/04-ART12
https://doi.org/10.1145/3629981

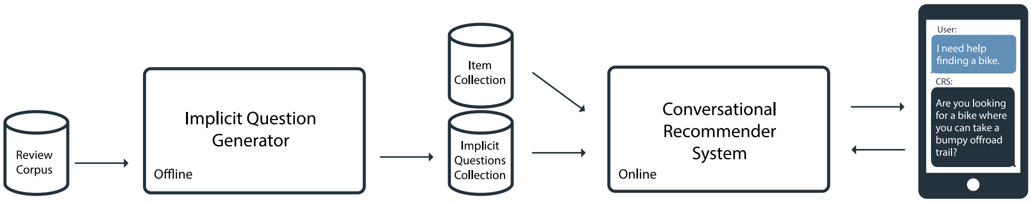

Review Corpus
Implicit Question Generator
Offline
Item Collection
Implicit Questions Collection
Conversational Recommender System
Online
User:
I need help finding a bike.
CRS:
Are you looking for a bike where you can take a bumpy offroad trail?

Fig. 1. Conceptual system overview. Our focus in this article is on the implicit question generator component.

systems often do not take into account that users might have made mistakes in the past (e.g., regarding purchases) [66] or that their preferences change over time [20]. Additionally, for some users, there is little historical data which makes modeling their preferences difficult [26]. A ***conversational recommender system*** (**CRS**), on the other hand, is a multi-turn, interactive recommender system that can elicit user preferences in real-time using natural language [21]. Given its interactive nature, it is capable of modeling dynamic user preferences and taking actions based on users current needs [14].

One of the main tasks of a conversational recommender system is to elicit preferences from users. This is traditionally done by asking questions either about items directly or item attributes [9, 11, 12, 14, 28, 58, 59, 69, 74]. Asking people to review individual recommendations to establish the characteristics of a single item they need, especially in a domain that they are not expert in, is particularly time-consuming; therefore, the research is commonly focused on the estimation and utilization of users preferences towards attributes [14]. Common to these approaches is that the user is explicitly asked about the desired values for a specific product attribute, much in the spirit of slot-filling dialogue systems [15]. For example, in the context of looking for a bicycle recommendation, we might have wheel dimensions or the number of gears as attributes in our item collection. In this case, a system might want to ask a question like *"How thick should the tires be?"* or *"How many gears should the bike have?"* However, ordinary users often do not possess this kind of attribute understanding, which might require extensive domain-specific knowledge. Instead, they only know where or how they intend to use the item. For example, a user might only be interested in using this bike for commuting but does not know what attributes might be good for that purpose. The novel research objective of this work is to generate *implicit* attribute questions for eliciting user preferences, related to the intended use of items. This stands in contrast to explicit questions that ask about specific item attributes.

Our approach hinges on the observation that usage-related experiences are often captured in item reviews. By identifying review sentences that discuss particular item features or aspects (e.g., *"fat tires"*) that matter in the context of various activities or usage scenarios (e.g., *"for conquering tough terrain"*), those sentences can then be turned into preference elicitation questions. In our envisaged scenario, a large collection of implicit preference elicitation questions is generated offline, and then utilized later in real-time interactions by a CRS; see Figure 1 for an illustration.

In this article, our focus is on the offline question generation part, whereas the actual item recommendation is left as a separate, downstream task to be addressed. A main challenge associated with the question generation task is the collection of high-quality training data. As our first contribution, we address the problem of creating a sentence-to-question dataset by developing a multi-stage data generation protocol. It starts with *candidate sentence selection*, which can be automated effectively based on part-of-speech tagging and simple linguistic patterns. Then, we employ a multi-step manual data annotation process via crowdsourcing, which involves (1) question generation (given an input sentence, turn it into a question, if possible), (2) question validation (filtering the responses collected in the previous step), and (3) expanding question variety (producing paraphrased versions of the input questions). As our second contribution, we propose four *question generation* models that, given a review as input, produce an implicit question in an end-to-end fashion. The simplest, template-based model uses the most common n-grams found in the training data to construct questions. The second model extends this template-based baseline by adding a classifier to discard non-applicable sentences before generating a question. The last two are neural models we fine-tuned for this particular task, from a pre-trained checkpoint of a sequence-to-sequence model for text generation [51]. The difference between the latter two lies in what is taken as the input—the first model uses heuristically extracted sentences, while the second one uses an entire review. The evaluation of our proposed approach is done against held-back test data using standard metrics for *automatic evaluation* of text generation (BLEU, ROUGE, and METEOR). Additionally, we evaluate the task in terms of Accuracy, i.e., whether a question can be constructed based on the given input. In *human evaluation,* we measure the effectiveness and the capability of our model to generate questions that are suitable for preference elicitation, can be answered easily, and are grammatically correct. The evaluation is performed both in pointwise and pairwise fashion, using a 5-point Likert scale. We find that all evaluations results (both automatic and two flavors of human evaluation) point to the same conclusions: that our proposed neural models outperform the strong template-based baseline. There are advantages to both neural models: the sentence-based model generates questions of slightly higher quality, while the review-based one has the advantage of being an end-to-end model with a simpler architecture.

In summary, our main contributions in this article are as follows:

- Introduce the novel task of eliciting preferences in CRSs via usage-related questions.
- Develop a multi-stage data annotation protocol using crowdsourcing for collecting high-quality ground truth data.
- Introduce two template-based and two neural approaches for generating usage-related questions based on a corpus of item reviews.
- Develop human evaluation protocols, conduct both automatic and manual evaluation of the proposed approaches, and perform an extensive analysis of results.

The resources developed in this article (crowdsourced dataset and question generation model) are made publicly available at https://github.com/iai-group/tors2023-crs-questions.

2 RELATED WORK

The focus of this work is preference elicitation via natural language in the context of conversational recommender systems. In this section, we discuss related work on conversational recommender systems, preference elicitation, and question generation.

2.1 Conversational Recommender Systems

Static recommendation models predict users' preferences based on their previous interactions with the system. Some of the more common early approaches include **collaborative filtering** (**CF**) [56], **logistic regression** (**LR**) [45], and **gradient boosting decision tree** (**GBDT**) [5]. The availability of datasets on user behavior data (e.g., click history, visit logs, ratings on items[1, 2]) has inspired, in recent years, the development of more sophisticated neural models such as **neural factorization machines** (**NFM**) [17] or **graph convolutional networks** (**GCN**) [72]. A significant drawback of static recommenders is that they treat recommendation as a *one-shot* interaction process under the assumption that the user's preferences lie in historical data. However, this does not hold in

[1] https://grouplens.org/datasets/movielens/
[2] https://www.baltrunas.info/context-aware

cases where there are no past observations [26]. This is often the case in scenarios where the user has not interacted with the system (cold-start problem) or in the case with high-involvement products (i.e., products that customers do not buy frequently and tend to invest more time and effort when selecting them) [21]. Wang et al. [66] note that data on clicks and purchases could be misleading, because a large portion of clicks do not lead to purchases, and when they do, users might have regretted their choice. Furthermore, the user's preferences might change over time [20] and capturing their past interactions can lead to recommendations that are no longer relevant. To deal with short-term but dynamic preferences, *session-based recommenders* have emerged and received considerable attention in recent years [65]. These algorithms provide recommendations solely based on the user's interactions during a continuous period of time (i.e., a session).

A CRS helps users reach their recommendation-oriented goals via multi-turn conversation [21]. While they share the goal of recommending items to users with traditional, static recommender systems, they do so by eliciting the detailed and current user preferences interactively in real-time. In contrast to session-based recommenders, where user preferences are implicit and inferred from interactions, users explicitly express their preferences here using natural language. Additionally, a CRS can provide explanations for the suggested items and process user feedback on the recommendation. While there are many open issues around CRSs, Gao et al. [14] identified the following five as primary challenges:

- *Question-based User Preference Elicitation.* The challenge is to generate questions that elicit as much information as possible and to use the provided information to make better recommendations. Two main lines of research are item-based [12, 58, 77] and attribute-based preference elicitation [27, 74]. Both approaches try to answer the questions of what to ask and how to adjust the recommendation based on user response.
- *Multi-turn Conversational Recommendation Strategies.* The main challenge is to balance continued question asking to reduce preference uncertainty and provide recommendations using the least number of conversation turns.
- *Natural Language Understanding and Generation.* One of the hardest challenges in CRSs is to communicate like a human [14]. Commonly, this involves providing a recommendation list directly or incorporating recommended items in a rule-based natural language template [15, 16, 73]. Recently, end-to-end frameworks have been proposed to understand users' intents and generate readable, fluent, and meaningful natural language responses [30].
- *Trade-offs between Exploration and Exploitation.* The dynamic nature of CRSs allows them to actively explore unseen items to capture user preferences. However, users generally have limited time and energy to interact with the system, therefore systems need to balance exploration with exploitation to make accurate recommendation.
- *Evaluation and User Simulation.* The complexity of evaluating CRSs comes from the emphasis on user experience during interactions. Systems need to be evaluated both on the turn and on the conversation level. While static recommenders can utilize large quantities of historical data to evaluate models, obtaining large number of user interactions to evaluate CRS is expensive [19]. Therefore, user simulation-based evaluation has been identified as a promising direction [73, 78].

In this article, we focus on question-based user preference elicitation and natural language generation. That is, we provide novel answers to questions *what to ask* and *how to ask*.

2.2 Preference Elicitation

Commonly, preference elicitation questions target either items or their attributes. Typical of early studies on CRSs, *item-based elicitation* approaches to ask for users' opinions on an item itself, using a combination of methods from traditional recommender systems, such as collaborative filtering, with user interaction in real time [64, 77]. These systems continuously recommend items and refine the recommendations based on user feedback. In case of *choice-based methods*, users are presented with two or more items. In every turn, the recommendation is updated based on the selected choice. The selection of items may be approached as an optimization problem using a static preference questionnaire method [58]. Another line of research is using probabilistic, multi-armed bandit algorithms that maximize the cumulative expected reward over some fixed number of rounds. There is an inherent exploration-exploitation tradeoff in these systems where exploration refers to acquiring information about arms, while exploitation is optimizing for the immediate reward in the current round [12, 64]. This method has a natural setup in the CRS setting where items can be seen as arms and rounds as the conversation turns.

Asking about items directly can be inefficient, as large item sets would require several conversational turns and in turn increase the likelihood of users losing interest [14]. Alternatively, *attribute-based elicitation* aims at predicting the next attribute to ask about. It is often cast as a sequence-to-sequence prediction problem, lending itself naturally to sequential neural networks [10, 18]. There has been an effort to create large datasets consisting of human conversations that can be used as training data. However, non-conversational data is often leveraged, especially when there is a lack of relevant information in the recorded dialogues [21]. Christakopoulou et al. [11] propose a **question & recommendation** (**Q&R**) method, utilize data from a non-conversational recommendation system, and develop surrogate tasks to answer questions: *What to ask?* and *How to respond?* To answer the first question, they develop a surrogate task where the goal is to predict the next likely topic a user would be interested in, based on recently watched videos. The second question is answered by predicting what video the user would be most interested in, based on the most relevant predicted topic. A similar approach of training a sequential neural network on non-conversational data is taken by Zhang et al. [74], who convert Amazon reviews into artificial conversations. Sentences with aspect-value pairs are extracted from reviews and serve as utterances in one round of conversation. The extracted aspect-value pairs are modeled as user information needs. The assumption is that the earlier aspect-value pairs appear in the review, the more important they are to the user, and thus should be prioritized as questions. Additionally, they develop a heuristic trigger to decide whether the model should ask about another attribute or recommend an item. The drawback of these systems is they have no way of modeling the rejection of recommendations by the user, since the goal is to fit historical data as it happened. Furthermore, it is not possible to determine the reason behind the user interaction, i.e., why the user chose that particular item [14].

Another way to elicit preferences is in the form of *critiques*, i.e., feedback on attribute values of recommended items [9]. For example, if the recommendation is for a *phone*, a critique might be *"not so big"* or *"something cheaper."* Such methods often employ heuristics as elicitation tactics [38, 39]. In recent work, Balog et al. [3] study the problem of robustly interpreting unconstrained natural language feedback on attributes. Our work differs from prior efforts in that we do not ask about specific attribute values directly, but instead ask indirect questions related to the planned use of an item.

To help interactively search and navigate the space of item, *facet-based selection* is a commonly used interaction paradigm, especially in e-commerce [60]. Facets correspond to a particular way of grouping items, based on attribute-value combinations. For a given item category, facets may be identified by domain experts or sorted dynamically in order to allow for a quick drill-down of the results [62]. Our work may be seen as a different way of clustering items, around item usage. However, different from facet selection, there is no linear constraint on a single facet—item usage maps to a subset of the attribute space, without the user necessarily knowing what the facets are. In practice, item selection often involves balancing a tradeoff, e.g., a bike that is practical for daily usage and can be taken off-road occasionally. This type of selection can be done based on usage, but not with facets/attributes.

2.3 Question Generation

While there is research on end-to-end frameworks to enable CRSs to both understand user intentions as well as generate fluent and meaningful natural language responses [30], the predominant approach is still to use templates or construct the utterances using predefined language patterns [14]. In recent years, the broader field of dialogue systems has brought forth two additional strands of research applicable to CRSs as well: retrieval-based and generation-based methods [42]. Instead of relying on a handful of templates, *retrieval-based methods* utilize a large collection of possible responses. The basic approach to retrieving the appropriate response is based on some notion of similarity between the user query and candidate responses, with the simplest being inner product [70]. *Generation-based methods* in dialogue systems are typically based on sequence-to-sequence modeling. These models are usually trained on a hand-labeled corpus of task-oriented dialogue [6]. Due to the limited amount of training data, *delexicalization* is used to increase the generality of the systems. It is the process of disassociating specific words from the lexicon by replacing them in the training set with generic placeholders. The sequence-to-sequence model is then trained to produce a delexicalized sentence (utterance skeleton) as output. To get the final sentence, the output utterance is *relexicalized* based on user need [22]. Although retrieval-based approaches have been explored to a lesser extent than generation-based methods, their potential to leverage large, existing dialogue datasets to provide contextually relevant and high-quality responses has been demonstrated, resulting in an improved conversational user experience [42, 43]. Our proposed approach shares elements of both of retrieval-based and generation-based methods: it generates questions using a sequence-to-sequence model and stores them in a collection that can be queried using retrieval-based methods. However, the task we focus on is fundamentally different. Namely, we are concerned with preference elicitation through the generation of implicit questions based on item usage, rather than simply responding to user queries or generating dialogue. This renders existing approaches inadequate for our task.

The problem of preference elicitation is also related to that of clarification of information needs in information-seeking scenarios. When searching for information, user queries are often ambiguous, faceted, or incomplete. To improve the user satisfaction, systems may decide not to provide an answer (e.g., based on their estimated confidence in the results) but instead proactively ask the user questions to clarify their needs [1]. This is especially important in conversational information seeking scenarios, where the system can return only a limited number of results due to the limited bandwith user interface. Similar to research in CRS, existing approaches to generating clarifying questions include retrieval-based methods [1, 52, 71] and generation-based methods [53, 67]. Our work differs from this line of work in that instead of clarifying an already expressed need, we are trying to elicit a new user information need.

2.4 Sequence-to-Sequence Modeling

The task of sequence-to-sequence models is to generate a sequence of output tokens conditioned on the input sequence. To generate high-quality output, transfer learning has proved to be a powerful technique. In transfer learning, a model is first pre-trained on a data-rich task, then fine-tuned on a downstream task. Early implementations used recurrent neural networks [48], however, in recent years, the Transformer architecture is more commonly used [63]. Within the Transformer framework, three main variants emerged: encoder-only, decoder-only, and encoder-decoder models. Encoder-only models, like BERT [13], are mainly used for classification. On the other hand, for text generation tasks, decoder-only [50] and encoder-decoder models [29, 51] are often employed.

One of the main differences of the two variants used for generation, apart from the architecture, is in the pre-training regime. Encoder-decoder models are generally pre-trained using causal masked token prediction, where a number of tokens in any position of the input sequence are masked and the model predicts the masked tokens based on the context. Decoder models, on the other hand, are pre-trained using a next token prediction strategy based on the input sequence plus the tokens predicted thus far. Both training regimes are conducted in an unsupervised fashion, and the goal is to learn language syntax and semantics, and store that information in the model weights. In this work, we apply sequence-to-sequence models to the question generation task with the goal of generating usage-related questions using different inputs (sentences or entire reviews) as context.

3 APPROACH

Our objective is to understand users' needs with minimal cognitive effort on their part. To overcome the shortcomings associated with item-based elicitation (large item space and slow narrowing of the recommendation candidates) and attribute-based elicitation (domain-specific knowledge required), we propose asking usage-related questions instead. These should be easier for users to answer and can thus lead to a better conversational user experience.

As a first step toward that objective, in this work, we focus on the generation of implicit elicitation questions—implicit in the sense that we ask users about the intended use of items as opposed to soliciting the values of specific attributes. To generate usage-related questions, we leverage review corpora under the assumption that reviewers bring attention to item usage, where or how an item was used, and whether or not it was suitable for the intended purpose. We want to identify item uses that occur sufficiently frequently and could be converted to a good question to present to a new user. Item review datasets tend to be very large, with both the number of items and reviews in the thousands or even millions, making manual labeling the entire dataset extremely expensive [31]. To overcome this, we develop automated approaches that can take a review as input and generate one or multiple preference elicitation questions out of that, if it is possible. We present multiple methods with an increasing degree of automation:

- We start with a *template-based* baseline approach that follows a pipeline of steps: first splitting reviews to sentences, then selecting sentences that mention some item-related activity or usage, and finally turning those sentences to questions using a pre-defined pattern (Section 3.1). This approach will always yield a question if the input sentence mentions an activity.
- Our second baseline extends the template-based approach by adding a classifier that is tasked with selecting only those sentences that could be converted to good questions. Otherwise, it still uses templates to construct questions from the selected sentences (Section 3.2).
- Next, we introduce a *neural sentence-based* approach, which still operates on the sentence level, but handles activity detection and question generation in an end-to-end manner using a large pre-trained language model (Section 3.3).
- Finally, we present a *neural review-based* method, which takes an entire review as input and generates a review questions from that, if it is possible (Section 3.4).

Figures 2–4 present schematic overviews of the different methods.

3.1 Baseline 1: Template-based Question Generation

Figure 2 illustrates the components of our ***template-based question generation*** (**TQG**) approach.

3.1.1 Candidate Sentence Selection. We identify sentences that describe some item feature or aspect and mention some activity or usage. For example:

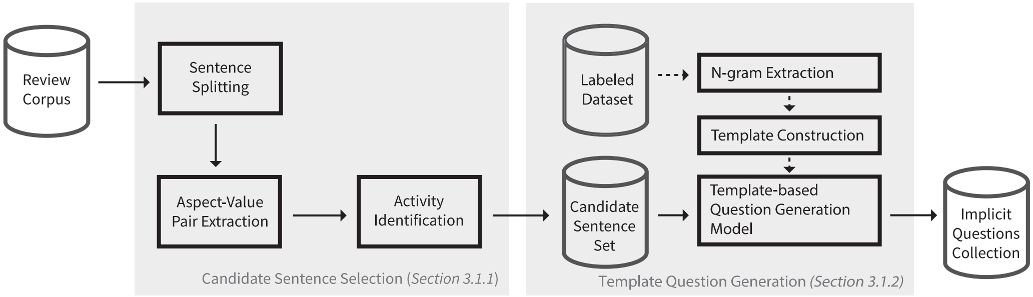

Review Corpus
Sentence Splitting
Aspect-Value Pair Extraction
Activity Identification
Candidate Sentence Selection (Section 3.1.1)
Labeled Dataset
N-gram Extraction
Template Construction
Candidate Sentence Set
Template-based Question Generation Model
Template Question Generation (Section 3.1.2)
Implicit Questions Collection

Fig. 2. Components of our template-based question generation system.

$$\texttt{The}\ \overbrace{\texttt{fat}}^{value}\ \overbrace{\texttt{tires}}^{aspect}\ \texttt{are perfect for}\ \overbrace{\texttt{conquering tough terrain}}^{usage/activity}\texttt{.}$$

Aspect-Value Pair Extraction. An aspect in this context is a term that characterizes a particular feature of an item [37] (e.g., *wheel*, *seat,* or *gear* are aspects of a bicycle). Value words are terms that describe an aspect (e.g., a *wheel* might be *large* or *small*, a *seat* can be *hard* or *comfortable*). Here, we extract all sentences that mention some aspect-value pair for a given category of items, using phrase-level sentiment analysis proposed by Zhang et al. [75, 76]. The motivation for this step stems from the assumption that an activity or usage can be mapped to a particular aspect of an item.[3]

Activity Identification. In this step, the goal is to classify sentences that mention some item-related activity or usage. Inspired by Benetka et al. [4], our approach revolves around using ***part-of-speech*** (**POS**) analysis and rules of the English language. We filter for the preposition *for* followed by a verb in progressive tense heuristically, by looking for *-ing* endings (e.g., *for commuting*, *for hiking*). This choice is driven by our intuition and was verified by manually inspecting a sample of the data. Note that there might be other formulations that describe activity or usage. Our goal is not to extract all possible sentences containing mentions of activity or usage; a high recall approach would likely come at the cost of a larger fraction of false positives. Instead, we focus on achieving high precision.

3.1.2 Question Generation. The main motivation for this step is generating natural-sounding questions that are easy for users to understand and answer, without needing any additional context. Consider the sentence *"The fat tires are perfect for conquering tough terrain."* An example of converting it to a yes or no usage-related question might be *"Would you like a bike that is perfect for conquering tough terrain?"* We approach this task using a template-based method, which is a common approach in CRS question generation [15, 16, 73].

There are many possible ways of articulating questions. To ensure that they are as natural-sounding as possible, we develop our template based on actual questions that humans formulated from review sentences. That is, we assume the presence of a training dataset consisting of sentence-question pairs, and inspect the most commonly appearing n-grams from the questions in that dataset. Specifically, in our training dataset (cf. Section 4), we observe the following as the most frequent question pattern:

```
Are you looking for a [category] that is great for [usage]?
```

[3] This concerns future utilization of responses given to these elicitation questions, where the CRS might want to map activities to specific attribute values.

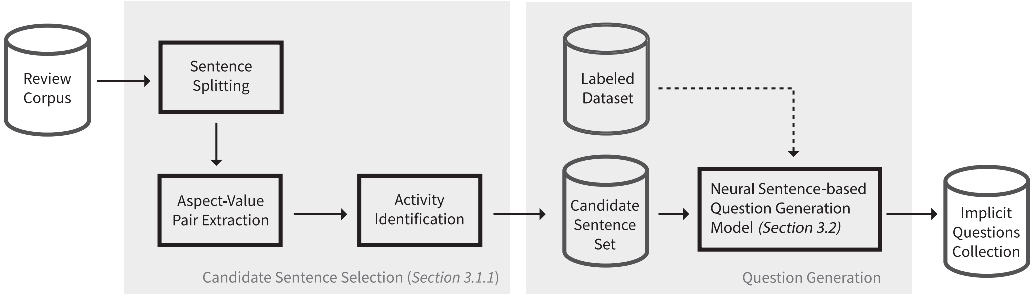

Review Corpus
Sentence Splitting
Aspect-Value Pair Extraction
Activity Identification
Candidate Sentence Selection (Section 3.1.1)
Labeled Dataset
Candidate Sentence Set
Neural Sentence-based Question Generation Model (Section 3.2)
Question Generation
Implicit Questions Collection

Fig. 3. Components of our neural sentence-based question generation system. The approach is similar to that of the template-based question generation, but instead of creating rigid templates, the model learns question patterns from the entire dataset automatically using a neural model.

An example question, based on this template, would be: *"Are you looking for a bike that is great for commuting?"*

Note that not all candidate sentences that pass our selection heuristic are viable for conversion to a question, e.g., *"Thank you so much for coming up with such a great product."* This sentence is too vague and does not mention any action or usage for the item, and thus should be labeled as **not applicable** (**N/A**). However, the simple template-based approach is not capable of making such a distinction and would generate a question regardless.

3.2 Baseline 2: Template-based Question Generation with Classification

With our second model (TQG+CLS), we address some of the limitations of the first baseline model. Specifically, we aim at avoiding generating questions that would not help with recommendations, because they would either be trivially answered affirmatively or would not make it easier to make a recommendation. Before generating a question, we classify the sentence as applicable or not applicable. If the sentence is not applicable, we do not generate a question. To achieve this, we train a transformer-based classifier to predict whether a sentence is applicable or not. We choose RoBERTa [35], a high-performant BERT-based transformer model. The input to the model is

```
input_seq = <cls> [sentence] <eos>,
```

where `<cls>` and `<eos>` are special tokens that mark the beginning and end of the sequence, respectively. `cls` is used as an input to a simple linear and a softmax layer, whose output gives us probability distribution over the two classes: applicable and not applicable.

3.3 Neural Sentence-based Question Generation

In our third model, ***neural sentence-based question generation*** (**NSQG**), depicted in Figure 3, we further address some of the limitations of the template-based approach. First, similarly to Baseline 2, we want to produce relevant questions for recommendations by avoiding those that are either easily answered affirmatively or do not contribute to facilitating a recommendation. For example, instead of generating the question *"Do you want a grill that is good for grilling certain things?,"* we want the model to output a special `[N/A]` (not applicable) token. Second, we would want to generate a richer variety of natural-sounding questions.

Learning to generate questions is done by fine-tuning a large, pre-trained, sequence-to-sequence language model. There are two main benefits of using transfer learning from a pre-trained model. First, it increases the learning speed; as both syntax and semantics of the English language are already learned, there are fewer things the model needs to learn. Second, it reduces the amount

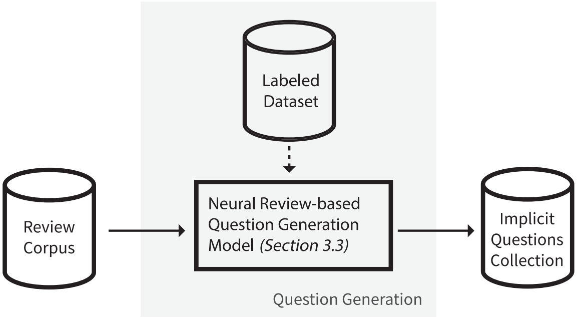


Fig. 4. Components of our neural review-based question generation system. The model drastically simplifies inference as we do not rely on heuristics to extract candidate sentences, but take entire reviews as input to generate questions.

of labeled data needed to train models to high performance. Specifically, we use T5 [51] as it has shown competitive performance across a variety of language generation tasks (e.g., conversational query rewriting [33] and document re-ranking [49]), and can be used for both N/A-classification and generation, where N/A-classification is posed as a text-to-text problem. We form the input to the T5 model as follows:

```
input_seq = Ask question: [category] <sep> [sentence] <eos>,
```

where "`Ask question:`" is a task-specific prefix, and `<sep>` and `<eos>` are the separation and end-of-sequence tokens, respectively. Considering that T5 was pre-trained on various tasks, a task-specific prefix is used to specify which task the model should perform. The output of the model is either a question or the `[N/A]` token.

We employ state-of-the-art techniques when performing model inference. Specifically, we use temperature-controlled stochastic sampling with top-$k = 25$ and top-$p = 0.90$ (nucleus) filtering. Top-k sampling restricts the sampling to consider only the 25 most likely next tokens. However, since some distributions from which tokens are sampled are flat while others are sharp, fixed k sampling is not optimal. To mitigate this shortcoming, nucleus filtering restricts the number of considered tokens to the minimum number of tokens whose total probability sums to p.

3.4 Neural Review-based Question Generation

The main motivation behind our last model, NRQG, is to simplify the process of question generation. The model is an extension of the previous sentence-based question generation (NSQG) model, except the input being an entire review instead of a single sentence.

On a high level, review-based question generation is a two-step process: a text extraction step, to identify a review sentence, followed by a text generation step where the sentence is "translated" into a question. That is, meaningful usage-related information first needs to be found in a longer text and then converted into a question. While sentence-based approaches use a heuristic for the first step and either a template (TQG) or a trained neural model (NSQG) for the second, with NRQG we simplify the pipeline considerably. Using a neural architecture allows us to perform both steps jointly by fine-tuning a large pre-trained language model in an end-to-end fashion, as illustrated in Figure 4.

The input to the NRQG model follows a similar structure to NSQG, except we replace the sentence with a review.

```
input_seq = Ask question: [category] <sep> [review] <eos>.
```

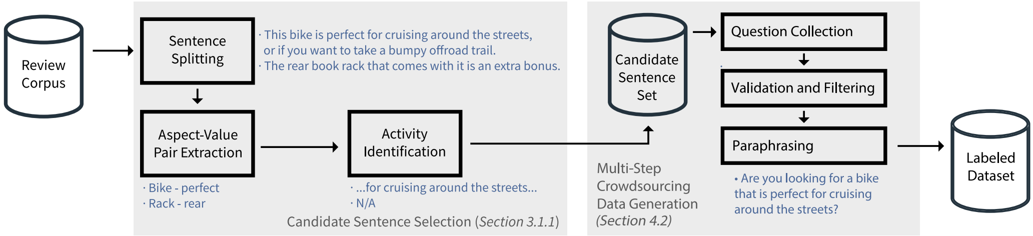

Review Corpus
Sentence Splitting
· This bike is perfect for cruising around the streets, or if you want to take a bumpy offroad trail.
· The rear book rack that comes with it is an extra bonus.
Aspect-Value Pair Extraction
· Bike - perfect
· Rack - rear
Activity Identification
· ...for cruising around the streets...
· N/A
Candidate Sentence Selection (Section 3.1.1)
Candidate Sentence Set
Multi-Step Crowdsourcing Data Generation (Section 4.2)
Question Collection
Validation and Filtering
Paraphrasing
• Are you looking for a bike that is perfect for cruising around the streets?
Labeled Dataset

Fig. 5. Data collection pipeline, consisting of automatic candidate sentence extraction based on linguistic patterns and multi-step manual data annotation via crowdsourcing.

While many neural models have an input limitation, usually of 512 tokens, T5 has no such limitation. However, long input sequences drastically slow down generation, and the common practice is to avoid them. In our experiments, only a handful of reviews were longer than 512 tokens with the longest having 2,000 tokens. If long reviews were common, a solution to processing longer reviews would be to use the same approach by splitting the reviews into manageable-sized chunks. Another consideration is dealing with reviews that mention multiple possible uses for an item. This is not a common scenario, and there are indeed no examples of such cases in our dataset. Therefore, we make the simplifying assumption that at most a single question may be generated from one review.

It is important to note that while we use an existing model for text generation in both sentence-based and review-based models, obtaining high-quality labeled data for fine-tuning the model is a challenge on its own. As one of the contributions of this article, we develop a multi-step data collection protocol using crowdsourcing, which we discuss in Section 4.2. Furthermore, while NRQG simplifies the modeling part considerably, it still relies on high-quality training data. The candidate sentence selections described in Section 4.1 is thus instrumental to facilitating data collection. In our experiments (in Section 6), we will analyze the impact of the pre-trained language model (i.e., number of parameters) as well as the amount of training data available for fine-tuning.

4 DATA COLLECTION

This section describes the process of creating our dataset, which consists of a set of review sentences and either (i) a corresponding set of five preference elicitation questions or (ii) the label N/A for each. The data collection pipeline is shown in Figure 5.

4.1 Candidate Sentence Selection

The starting point for getting the candidate sentences are the Amazon review and metadata datasets [46],[4] where item reviews from Amazon are extracted along with product metadata information such as *title*, *description*, *price*, and *categories*. There are, in total, 233.1 M reviews about 15.5 M products. Due to the sheer dataset size, we focus our research on three main categories: *Home and Kitchen*; *Patio, Lawn and Garden*; and *Sports and Outdoors*. From these (40 M reviews), we further sub-select 12 diverse subcategories (referred to as *categories* henceforth): *Backpacking Packs*, *Tents*, *Bikes*, *Jackets*, *Vacuums*, *Blenders*, *Espresso Machines*, *Grills*, *Walk-Behind Lawn Mowers*, *Birdhouses*, *Feeders*, and *Snow Shovels*. This narrowed down the number of reviews to 989 k.

Sentence splitting and *aspect-value* pair extraction is performed using the Sentires toolkit [75, 76].[5] This step discards many non-viable sentences. The remaining ones are POS-tagged using

[4] https://nijianmo.github.io/amazon/index.html
[5] https://github.com/evison/Sentires

the Stanford NLP toolkit [41]. Finally, we filter for sentences that match our activity detection heuristic (*"for + [verb in progressive tense]"*). After this step, we were left with 14,140 reviews. Our sentence selection process is designed to favor precision over recall, and was validated by manual inspection of the results. Upon completion of the crowdsourcing tasks (described in Section 4.2), we find that over 75% of the selected sentences can be turned into questions. This shows that our simple method can indeed identify candidate sentences with relatively high precision.

Our final *candidate sentence set* contains approximately 100 sentences per category. An exception is the *Birdhouses* category, where only 15 candidate sentences are found due to the size of that category. In total, the candidate set consists of 1,098 sentences over 12 categories.

4.2 Question Generation using Crowdsourcing

Crowdsourcing was done on the **Amazon Mechanical Turk** (**AMT**) platform in three steps. The task was available to workers with 95% approval rate and with at least 1,000 approved **human intelligence tasks** (**HITs**).

4.2.1 Step 1: Question Collection. Crowd workers are given a review sentence (describing some aspect or use for a product) and a product category as input, and tasked with rewriting it into a question or marking it as not applicable. They are specifically instructed to formulate a question that a salesperson or a recommender agent might ask a customer, such that it is a standalone question that can simply be answered with yes/no. For every input sentence, we collected responses from three different workers. Sentences found non-applicable by at least two workers are set as N/A. The task was re-run if a single worker responded with N/A. This process resulted in approximately 2,600 sentence-question pairs.

4.2.2 Step 2: Validation and Filtering. Next, we validate all responses (i.e., generated questions) for applicable sentences collected in Step 1 using crowdsourcing. We employ three different workers in Step 2, who are requested to answer four multiple-choice questions: (1) *Is the question grammatically correct?* [Yes/No] (2) *Can the question be answered by yes or no?* [Yes/No] (3) *Does the question mention any trait or use for a product?* [Yes/No] (4) *Who is most likely to ask this question in a sales setting?* [Buyer/Salesperson/Neither]. Generated questions that are found invalid by all three workers on a single aspect or at least two workers on at least two aspects are automatically rejected. Those that are marked invalid on multiple aspects but do not fall into the former category are manually checked by an expert annotator (one of the authors). All other questions are approved. Steps 1 and 2 were run multiple times until all questions were resolved.

4.2.3 Step 3: Expanding Question Variety. Our main motivation for expanding the question variety is to add new ways of asking implicit questions. To this end, we task a new set of workers to paraphrase the questions we obtained and validated in Steps 1 and 2. Each worker receives all three versions of the questions from Step 1 as input and is asked to produce a new (paraphrased) question that expresses the same meaning. Note that this set of workers do not get to see the original sentences, only the questions generated from them by other workers. For each set of three questions, two additional paraphrases were collected. Considering that generating paraphrases proved to be a much simpler task than generating questions from review sentences, no additional quality assurance steps were necessary.

4.3 Final Dataset

Out of the 1,115 candidate sentences, 277 were labeled as non-applicable (not containing relevant usage-related information), which is below 25%. This shows that our high-precision approach to selecting candidate sentences is effective. We note that our sentence selection method works better

Table 1. Example Sentence-question Pairs from our Dataset

Category	Blender
Sentence	Great for making smoothies with frozen fruit.
Generated questions	- Are you looking for a blender that's great for making smoothies with frozen fruit? - Would you be interested in a blender that is great for making smoothies with frozen fruit? - Are you interested in a blender for making smoothies with frozen fruit?
Paraphrases	- Do you want a blender that's great for making smoothies with frozen fruit? - Would you like a blender that is great for making smoothies with frozen fruit?

Category	Snow shovel
Sentence	This product is excellent for doing the job
Generated questions	n/a n/a n/a
Paraphrases	

for some categories than for others. The fraction of viable sentences ranges between 52% (*Espresso machine* category) to 84% (*Backpacking pack* category). For the remaining 838 sentences, a total of five questions are generated, three based on the candidate sentence and two via paraphrasing. Table 1 shows two example sentences from our dataset.

5 EXPERIMENTAL SETUP

This section presents the experimental setup for the three methods explored in this article. We evaluate our models using standard automatic metrics for evaluating text generation. We also perform human evaluation via a set of crowdsourcing studies to assess question quality across multiple dimensions.

5.1 Question Generation

For our neural approaches (NSQG and NRQG), we train *small*, *base*, and *large* T5 models, which vary in the number of layers, self-attention heads, and the dimension of the final feedforward layer. The difference is shown in the number of parameters in Table 5. We use 80% of the data for training, while the rest is test data. In our training, we employ teacher forcing [68], regularization by early stopping [44], and adaptive gradient method AdamW [36] with linear learning rate decay. For each sentence, we have either N/A or a set of reference questions as ground truth.

5.2 Automatic Evaluation

We evaluate question generation as a classification task in terms of Accuracy (detecting N/A), and as a machine translation task, where the set of human-generated questions serve as reference translations. Specifically, we report on BLEU-4, which uses modified n-gram precision up to 4-grams [47], and ROUGE-L, a recall-based metric based on the longest common subsequence [32]. Additionally, we report METEOR, which has been found to have a better correlation with human judgments compared to BLEU and ROUGE [25]. It does this by considering word stems, WordNet synonyms, and paraphrases in addition to n-gram overlap.

While evaluating sentence-based models (TQG and NSQG) is straightforward, there is a detail we have to consider when generating and evaluating questions using the review-based (NRQG) method. Each review may contain multiple sentences mentioning usage that could potentially be used to generate questions. However, in our dataset, we do not have any such instances (i.e., no two sentences happen to come from the same review). While this is not by design, intuitively it makes sense that people do not discuss multiple usages of an item within a single review. Therefore, we can evaluate the NRQG model exactly the same way we evaluate the TQG and NSQG models—that is, for each input review we expect a single generated question. The generated question is then compared to the ground truth questions in the dataset.

5.3 Human Evaluation

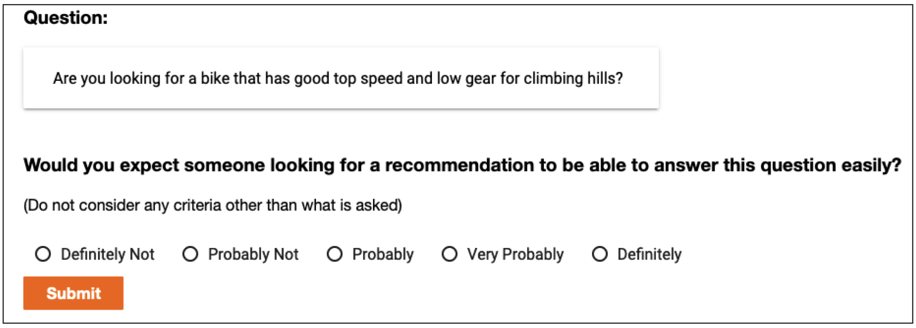
Recent studies have shown that automatic measures often have a low correlation with human judgement [7, 34, 40, 55, 57]. To thoroughly investigate the differences between our question generation models, we additionally evaluate them by human assessors. Human evaluation of natural language generation is most commonly done with respect to a single dimension, however, it has been observed that there are many aspects of language generation that cannot be captured in a single metric [61]. In our work, we consider three quality dimensions: grammar and fluency, usability, and answerability.

We compare the generated questions both on their own (*pointwise* evaluation) and relative to each other (*pairwise* evaluation) with the help of crowd workers. Research suggests that pairwise comparison might be more reliable [8], however, the cost of evaluation increases with the number of models. Another reason for performing pointwise evaluation is that it yields an *absolute* measure by averaging over a set of questions. Pairwise evaluation, on the other hand, can only establish a *relative* ordering between two approaches. Nevertheless, both absolute and relative measurements can be insightful and we are particularly interested in seeing if the observations we can draw from them are in alignment. In both cases, we focus on three different aspects of the questions, which in combination describe their *naturalness*.

An important aspect of conversational systems is to generate fluent, coherent, and grammatically correct utterances [2]. Therefore, the first evaluation dimension focuses on *grammar and fluency*. In pointwise evaluation, we ask *"Is the question fluent and grammatically correct?,"* while in the pairwise case, we ask *"Which question is more fluent and grammatically correct?"* Another important aspect when generating questions that are supposed to elicit user preferences is *usefulness*. Rosset et al. [54] introduce the concept of usefulness in conversational search and describe it as a measurement for how well a suggestion leads the user to useful information. In our case, we are trying to evaluate how useful the question is in making a good recommendation. In other words, does answering the question help with giving a better recommendation? We ask (pointwise) *"If you were making a recommendation for a friend, would knowing the answer be useful for you to make a better recommendation?"* or (pairwise) *"If you were making a recommendation for a friend, the answer to which question would be more useful for you to make a better recommendation?"* The final aspect we explore is that of *answerability*, i.e., how easy or difficult it is for the user to answer the question. Is the question ambiguous or straightforward? For example, a question *"Are you looking for a snow shovel that is extremely good snow shovel for Wyoming?"* might be easy to answer for most people living in Wyoming or if we know what the typical winter is like there. However, this kind of question is very specific and difficult to answer for most people outside Wyoming. In pointwise evaluation, we ask *"Would you expect someone looking for a recommendation to be able to answer this question easily?,"* while in pairwise evaluation, we ask *"Which question is easier to answer when looking for a recommendation?"*

In pointwise evaluation, we solicit answers on a 5-point Likert scale. An example can be seen in Figure 6 where the responses range from "definitely not" to "definitely." In pairwise evaluation, we also employ a 5-point scale, where the two ends of the spectrum correspond to strong preferences for each question, with gradually weaker preferences in between and "no preference" in the middle. An example pairwise evaluation task is shown in Figure 7.

We use crowdsourcing on Amazon Mechanical Turk to evaluate the generated questions in our study. Each question is annotated by three different workers, all based in the United States or Great



Question:

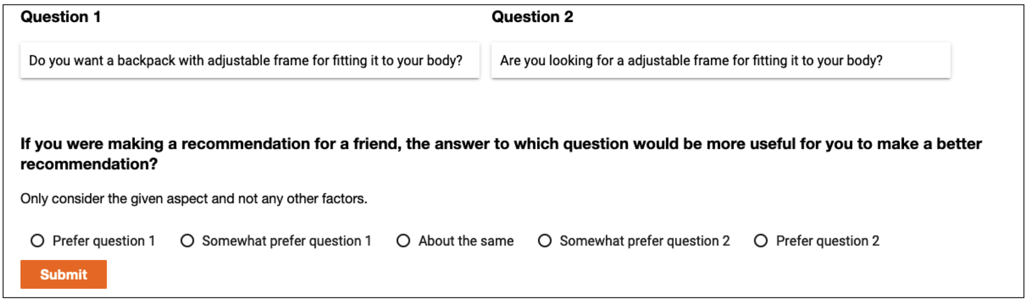
Are you looking for a bike that has good top speed and low gear for climbing hills?

Would you expect someone looking for a recommendation to be able to answer this question easily?

(Do not consider any criteria other than what is asked)

○ Definitely Not ○ Probably Not ○ Probably ○ Very Probably ○ Definitely

Fig. 6. Example question for pointwise evaluation. The specific task addresses the answerability of the question in the context of providing better recommendations.



Question 1

Do you want a backpack with adjustable frame for fitting it to your body?

Question 2

Are you looking for a adjustable frame for fitting it to your body?

If you were making a recommendation for a friend, the answer to which question would be more useful for you to make a better recommendation?

Only consider the given aspect and not any other factors.

○ Prefer question 1 ○ Somewhat prefer question 1 ○ About the same ○ Somewhat prefer question 2 ○ Prefer question 2

Fig. 7. Example question for pairwise evaluation. The specific task addresses the usefulness of the presented question.

Britain, with a minimum approval rate of 95%, and a minimum number of accepted HITs 1,000. We take the mean of the three annotations as the final score for each question.

6 RESULTS AND ANALYSIS

The main research question we wish to answer with our experiments is the following: *Given a review from a corpus, can item-usage questions for preference elicitation be automatically generated?* To address this question, we break the problem down into more specific research questions:

- **RQ1** Can neural models generate more natural questions when compared with template-based baselines?
- **RQ2** How does (a) the size of the pre-trained language model and (b) the volume of available training data affect the performance of the neural models?

Specifically, given a review as input, our approaches should either generate a question or label it as N/A if a usage-related question cannot be generated or where the generated question would not be useful.

6.1 Automatic Evaluation

First, we compare the neural models with the two baseline models using automatic evaluation (RQ1). We train *T5-large* for both NSQG and NRQG as it was found to be the most effective model for the task. The results are reported in Table 2. We find that both neural models significantly outperform the template-based models on all generation evaluation metrics. This is expected, as neural models are capable of using both syntax and semantics present in the original sentence

Table 2. Performance Comparison of Different Question Generation Models: Template-based (TQG), Neural Sentence-based (NSQG), and Neural Review-based (NRQG)

Model	N/A Accuracy	BLEU-4	ROUGE-L	METEOR
Baseline 1 (TQG)	0.728	0.604	0.723	0.418
Baseline 2 (TQG+CLS)	**0.870***	0.607	0.727	0.420
NSQG	0.858*	**0.730**$^{*+}$	**0.806**$^{*+}$	**0.494**$^{*+}$
NRQG	0.832*	0.684^{*+}	0.769^{*+}	0.466^{*+}

ll models utilize all available training data (i.e., five questions or N/A per sentence). The best scores for each measure are in boldface. The symbols * and $^{+}$ denote statistically significant improvements over the two baselines, respectively (p-value < 0.05). Statistical significance for accuracy is measured using McNemar's test, while for BLEU, ROUGE, and METEOR we use paired bootstrap resampling [23].

Table 3. Pointwise Evaluation of our Three Models for Each Quality Dimension, on a Scale of 1 to 5

Model	Grammar and Fluency	Usefulness	Answerability
Baseline 1 (TQG)	3.69	3.48	3.71
Baseline 2 (TQG+CLS)	3.85*	3.67*	3.86
NSQG	**4.02***	**3.81**$^{*+}$	**3.98***
NRQG	3.80	3.73*	3.93*

The best scores for each measure are in boldface. The symbols * and $^{+}$ denote statistically significance improvements over the two baselines, respectively (p-value < 0.05), measured using a non-parametric Mann-Whitney U test.

when generating questions. Unsurprisingly, they significantly outperform Baseline 1 on the classification task as well. Baseline 2 achieves the highest accuracy, suggesting that a dedicated classifier performs better than a general-purpose model on the classification task. We also observe that the sentence-based (NSQG) model outperforms the review-based one (NRQG) on all metrics. This is unsurprising as the NSQG model has a simpler task to perform, as it receives the already extracted candidate sentences as input. Note that the accuracy of the review-based model is much higher than that of the template-based model, and only slightly worse than that of the sentence-based model, which suggests that despite the larger (and arguably noisier) input, the model can predict with high accuracy if useful questions can be generated.

6.2 Human Evaluation

The questions generated by the four models are also evaluated using human assessors along three dimensions: grammar and fluency, usefulness, and answerability. The pointwise evaluation results are presented in Table 3. Overall, all models score above average (> 3) along all evaluation dimensions. The neural models outperform Baseline 1 when comparing *grammar and fluency*; the differences are significant for NSQG. This is expected as the characteristic property of using large pre-trained language models is their capability to use grammar correctly. When constructing templates using the most frequent n-grams, we have no guarantees of fluency or adherence to grammatical rules. However, it is interesting to note that grammar is adequate (i.e., scoring 3 or greater) in over 80% of the test cases and that Baseline 2 significantly improves it. In both *usefulness* and *answerability*, the neural models perform similarly, with the review-based (NRQG) being only slightly worse than the sentence-based (NSQG) model. They both significantly outperform Baseline 1, which likely follows from the fact that these models can accurately determine when not to generate a question and predict N/A instead.

Figure 8 shows the breakdown of the pointwise evaluation across all 12 product categories for the sentence-based neural model (NSQG). We see the scores are above average (i.e., above 3) for all categories on all three dimensions. Of the three dimensions, the scores for *grammar and fluency*

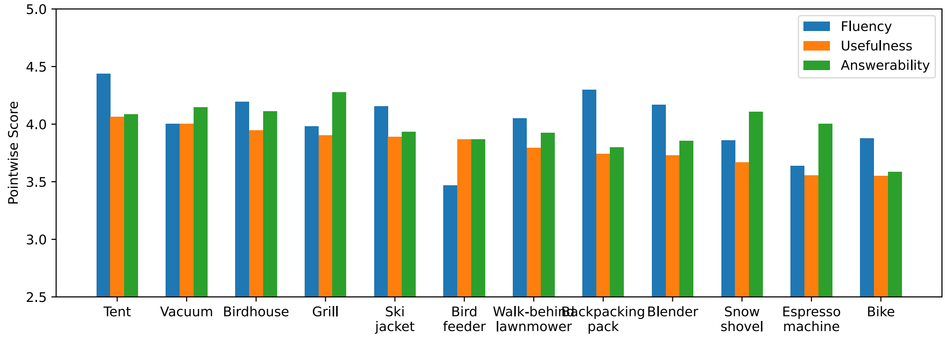


Fig. 8. Pointwise evaluation of the NSQG model per category. Categories are sorted by the usefulness score.

Table 4. Pairwise Evaluation According to Different Quality Dimensions

Grammar and fluency			
	Wins over		
	TQG	**NSQG**	**NRQG**
TQG	–	31% (10%)	38% (16%)
NSQG	44% (20%)	–	36% (15%)
NRQG	36% (17%)	28% (10%)	–

Usefulness			
	Wins over		
	TQG	**NSQG**	**NRQG**
TQG	–	22% (6%)	25% (7%)
NSQG	36% (17%)	–	25% (7%)
NRQG	36% (16%)	23% (7%)	–

Answerability			
	Wins over		
	TQG	**NSQG**	**NRQG**
TQG	–	32% (11%)	36% (16%)
NSQG	41% (18%)	–	34% (16%)
NRQG	39% (19%)	33% (14%)	–

The main values are percentages of how often the model in the row wins over the model in the column. The value in brackets are percentages of how often the model is strongly preferred.

are the highest overall, as well as for most categories. Interestingly, there is still a large variance between different categories, with the categories *Bird feeder* and *Espresso machine* having the lowest scores, and *Tent* and *Backpacking pack* highest scores. The categories *Bike*, *Espresso machine*, and *Snow shovel* have the lowest scores in terms of *usefulness*. It suggests that the model should label sentences as N/A more often for those categories.

The pairwise evaluation shown in Table 4 follows the same patterns as the pointwise evaluation. In all cases, annotators prefer the outputs of the NSQG model, followed by the NRQG model. The biggest distinction between the template-based and neural models is seen in *usefulness*, where the annotators prefer the neural models, often strongly so, in the vast majority of cases. There is almost no distinction between the neural models. However, NSQG is a clear favorite in the other two dimensions (i.e., *grammar and fluency* and *answerability*).

To answer our main research question, we conclude that overall, we can generate high-quality questions according to both automatic and human evaluation. Furthermore, based on human evaluation experiments, the neural models generate more natural questions compared to the template-based baselines (RQ1).

6.3 Model Size

Next, we explore what effect the size of the pre-trained language model has on the performance of neural question generation (RQ2a). Specifically, we fine-tune three T5 models of different sizes when employing neural sentence-based question generation (NSQG). Table 5 shows the results in terms of non-applicability classification (Accuracy) and question generation (BLEU, ROUGE, and METEOR). The model size does not have a large impact on the question generation task. The difference, however, is more pronounced for non-applicability (N/A) detection than for question generation. Detecting N/A is one of the most important parts of the pipeline since question

Table 5. Performance of the Sentence-based Question Generation (NSQG) Model using Different Pre-trained Language Models that are Fine-tuned on all Available Training data (i.e., Five Questions or N/A per Sentence)

Model	#Parameters	N/A Accuracy	BLEU-4	ROUGE-L	METEOR
T5-small	60.5 M	0.724	0.716	**0.810**	**0.497**
T5-base	222 M	0.819	0.693	0.794	0.493
T5-large	737 M	**0.858**	**0.730**	0.806	0.494

The best scores for each measure are in boldface.

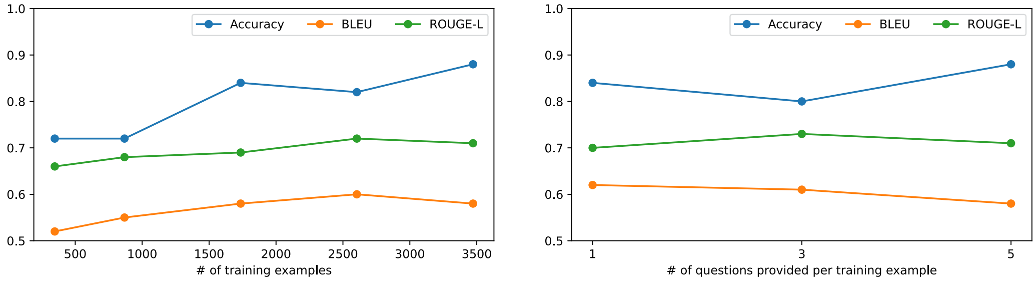


Fig. 9. Model performance (T5-large) with sentence-based (Left) or question-based (Right) training data reduction for the T5-large version of the NSQG model.

generation quality heavily depends on only converting useful item-usage sentences to questions. Furthermore, while there is a tradeoff between model size and accuracy, note that the planned usage is to generate questions offline and store them as a question collection. Thus, efficiency is not the main concern in this scenario. For this reason, we conclude that larger pre-trained models yield more effective questions.

6.4 Training Data Volume

We further investigate how the amount of training data affects model performance (RQ2b) by considering different ways and degrees of data reduction. As before, we use the best-performing NSQG model for this experiment, i.e., T5 large. In *sentence-based* data reduction, shown in Figure 9 (Left), only a subset of the available sentences is used for training (using all available questions corresponding to those sentences). We observe a drop in accuracy when we reduce the amount of training data to 25% or lower (i.e., less than 1,000 training samples), while question generation performance is less severely affected. In *question-based* data reduction, shown in Figure 9 (Right), we split the dataset based on the number of questions available for each sentence. We consider using a single question (1), the three initially generated questions (3), and the three initial questions plus the two paraphrases (5). We find that reducing the number of questions has surprisingly little effect. This suggests that it is more beneficial to collect a small number of questions for a larger set of sentences than vice versa.

6.5 Success/Failure Analysis

A closer look at specific sentence-question pairs reveals two patterns that leave room for future improvement; Table 6 shows examples of these using the template-based baseline models (TQG and TQG+CLS) and the best-performing variants of the neural models (NSQG and NRQG). We find that some of the generated questions are *too generic* (Table 6, top block). These are correct in terms of grammar and structure, but unsuitable for eliciting meaningful user preferences, e.g.,

Table 6. Examples of Question Generation Outputs for all four Models

Pattern	*Generic questions*
Ground truth	- n/a - n/a - n/a
TQG **TQG+CLS**	- Are you looking for a grill that is great for grilling certain things–not good for everything but what is? - Are you looking for a vacuum that is great for something to keep my floors clean – not ’eat off of’ clean? - Are you looking for a espresso machine that is great for making espresso drinks?
NSQG	- Do you need a grill that is good for grilling certain things? - Are you looking for a vacuum to clean your floors? - Do you want an espresso machine that is good for making espresso drinks?
NRQG	- Are you looking for a grill that is perfect for satay and quick grilling using smaller amounts of charcoal? - Are you looking for a vacuum cleaner that can keep your floors clean? - Are you looking for an espresso machine that is good for making espresso drinks?

Pattern	*Complex questions*
Ground truth	- n/a - Do you need a backpack with a good size for traveling on an airplane or going on a camping trip for a few days or packing for a few days trip?
TQG **TQG+CLS**	- Are you looking for a espresso machine that is great for something higher quality that’ll last longer and has more features or if you’d like an introductory machine that makes delicious espresso that’s much more affordable? - Are you looking for a backpacking pack that is great for traveling on an airplane (come up with ways to shrink it by removing the frame and using the straps) or going on a camping trip for a few days or packing for a few days trip?
NSQG	- Are you looking for an espresso machine that has a lot of features or if you’d rather spend a lot of money up front for something higher quality that’ll last longer and has more features? - Are you looking for a good size backpack that can be easily shrunk by removing frame and be used with straps?
NRQG	- Are you looking for an espresso machine that lasts longer and has more features? - Are you looking for a backpack that is a good size for traveling on an airplane or going on a camping trip for a few days or packing for a few days trip?

Pattern	*Success*
Ground truth	- Would you like a spacious backpacking pack? - Are you looking for a grill that you can take on camping trips,even the long ones? - Do you want a perfect tent for backpacking?
TQG **TQG+CLS**	- Are you looking for a backpacking pack that is great for everything i would need for a three day isolation and more? - Are you looking for a grill that is great for taking on camping trips (even long ones)? - Are you looking for a tent that is great for backpacking?
NSQG	- Are you in need of a backpack that has more than enough room for everything? - Are you looking for a grill that is perfect for long distance camping trips? - Are you looking for a tent that is perfect for backpacking?
NRQG	- Are you in need of a backpack that is in great shape and has more than enough room for everything? - Are you looking for a grill that is perfect for camping trips? - Are you looking for a tent that is perfect for backpacking?

“Do you need a grill that is good for grilling certain things?” Instead of returning N/A (which is indeed the corresponding response in our dataset), the model generated a question that is so vague and generic that it is hard to think of a scenario where it would not be answered affirmatively. Interestingly, the review-based model in this scenario utilized another part of the input instead

of the heuristically extracted sentence, which sentence-based models operate on, to generate a more useful question *"Are you looking for a grill that is perfect for satay and quick grilling using smaller amounts of charcoal?"* The second pattern concerns *complex questions* (Table 6, middle block) that ask about more than one usage or activity, e.g., *"Are you looking for a backpacking pack that is a good size for traveling on an airplane or going on a camping trip for a few days or packing for a few days trip?"* This question is too complex and unlikely to elicit any meaningful information without the user having to elaborate which options they agree with and which they do not. Such questions should instead be split into several simpler ones where it is both easier to interpret the question and to answer it. Note that crowd workers were not instructed to simplify complex questions, therefore it is not surprising that is what the model has learned. We also include examples of successes (Table 6, bottom block) where all three models generate valid questions. We notice that for shorter inputs, all three models generate useful and grammatically correct questions that are easy to answer. Since the template-based model is directly dependent on the structure of the input sentence, in some cases it does not produce a fluent question, e.g., *"Are you looking for a backpacking pack that is great for everything i would need for a three day isolation and more?"* However, the meaning is still understandable even if the usefulness is limited of such an over-specified question.

7 CONCLUSION AND FUTURE DIRECTIONS

In this article, we have studied the question of how a conversational recommender system can solicit user's needs through natural language by using indirect questions about how the wanted product will be used. This contrasts with most prior work that considers how to directly ask about desired product attributes. We have developed, evaluated, and compared four models used on the task: two template-based and two neural models. In each case, the start is a corpus of reviews, and the goal is to generate preference elicitation questions, if possible. We show that all four models effectively extract relevant information from reviews (with high precision), and transform it into useful questions. For the sentence-based models, sentences containing usage-related statements are identified heuristically, while the review-based model works end-to-end. The generated questions from all models are of high quality, with the neural models achieving higher scores in the automated evaluation and also being preferred by human annotators.

Utilization. We emphasize that this work focuses on this first stage of recommendation in a conversational setting, eliciting the user's needs in a natural and engaging way. The most important future direction is determining how answers to these questions should best be leveraged for the task of generating recommendations, once the user's need is understood. Here, we anticipate that sentence embedding techniques would likely to be effective. Second, as this work builds on top of large language models, language safety is a key consideration warranting further study before our approach could be used in practice. Nevertheless, during experimentation, we did not observe concerning language nor hallucinations. We also note that the offline question generation process lends itself well to even manual control over the language model output.

Limitations. We focus on generating high-quality questions (precision) as opposed the having an extensive coverage of the possible item uses (recall). We do not address the aspect of question diversity explicitly. Instead, it is assumed that human-created reviews naturally cover the different ways a given item is used. Determining whether the coverage of usage-related questions is sufficient for a given category would be an interesting direction for further investigation.

Generalizability. Our approach may be employed in other domains where items are associated with a certain activity. For instance, in a movie recommendation scenario, a statement like "This movie was perfect for watching with my kids," could provide valuable usage-related insights, as could similar statements in the domain of travel, food, or restaurants. Our approach may be less applicable in contexts where the items do not lend themselves to specific activities or usage scenarios, e.g., news recommendation.

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open–domain information–seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 475–484.

[2] Avishek Anand, Lawrence Cavedon, Hideo Joho, Mark Sanderson, and Benno Stein. 2020. Conversational search (dagstuhl seminar 19461). *Dagstuhl Reports* 9, 11 (2020), 34–83.

[3] Krisztian Balog, Filip Radlinski, and Alexandros Karatzoglou. 2021. On interpretation and measurement of soft attributes for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 890–899.

[4] Jan R. Benetka, John Krumm, and Paul N. Bennett. 2019. Understanding context for tasks and activities. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval.* 133–142.

[5] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. 2013. Entity recommendations in web search. In *Proceedings of the Semantic Web – ISWC 2013.* 33–48.

[6] Pawe\l Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* 5016–5026.

[7] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of bleu in machine translation research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics. 249–256.*

[8] Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Dumais. 2008. Here or there: Preference judgments for relevance. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval.* 16–27.

[9] Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 125–150.

[10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.* 1724–1734.

[11] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&R: A two–stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 139–148.

[12] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 815–824.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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). 4171–4186.*

[14] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open* 2 (2021), 100–126.

[15] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational AI. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1371–1374.

[16] Javeria Habib, Shuo Zhang, and Krisztian Balog. 2020. IAI MovieBot: A conversational movie recommender system. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management.* 3405–3408.

[17] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17).* 355–364.

[18] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[19] Jin Huang, Harrie Oosterhuis, Maarten de Rijke, and Herke van Hoof. 2020. Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems.* 190–199.

[20] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When people change their mind: Off–policy evaluation in non–stationary recommendation environments. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining.* 447–455.

[21] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2022. A survey on conversational recommender systems. *Computing Surveys* 54, 5 (2022), 1–36.

[22] Dan Jurafsky and James H. Martin. 2020. *Speech and Language Processing.* (3rd ed.). Draft. https://web.stanford.edu/~jurafsky/slp3/

[23] Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing.* 388–395.

[24] Ivica Kostric, Krisztian Balog, and Filip Radlinski. 2021. Soliciting user preferences in conversational recommender systems via usage-related questions. In *Proceedings of the 15th ACM Conference on Recommender Systems.* 724–729.

[25] Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the 2nd Workshop on Statistical Machine Translation.* 228–231.

[26] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta–learned user preference estimator for cold–start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1073–1082.

[27] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining.* 304–312.

[28] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat–Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2073–2083.

[29] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 7871–7880.

[30] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Proceedings of the Advances in Neural Information Processing Systems.*

[31] Yuan-Hong Liao, Amlan Kar, and Sanja Fidler. 2021. Towards Good Practices for Efficiently Annotating Large-Scale Image Classification Datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4350–4359.

[32] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Text Summarization Branches Out.* 74–81.

[33] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational Question Reformulation via Sequence-to-Sequence Architectures and Pretrained Language Models. arXiv:2004.01909. Retrieved from https://arxiv.org/abs/2004.01909

[34] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.* 2122–2132.

[35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692. Retrieved from https://arxiv.org/abs/1907.11692

[36] Ilya Loshchilov and Frank Hutter. 2022. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations.*

[37] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context–aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web.* 347–356.

[38] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent linear critiquing for conversational recommender systems. In *Proceedings of the Web Conference 2020.* 2535–2541.

[39] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep critiquing for VAE–based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 1269–1278.*

[40] François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* 1552–1561.

[41] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 55–60.*

[42] Ahtsham Manzoor and Dietmar Jannach. 2021. Generation-based vs. retrieval-based conversational recommendation: A user-centric comparison. In *Proceedings of the 15th ACM Conference on Recommender Systems.* 515–520.

[43] Ahtsham Manzoor and Dietmar Jannach. 2022. Towards retrieval-based conversational recommendation. *Information Systems* 109, C (2022), 102083.

[44] N. Morgan and H. Bourlard. 1989. Generalization and parameter estimation in feedforward nets: Some experiments. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems.* 630–637.

[45] J. A. Nelder and R. W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* 135, 3 (1972), 370–384.

[46] Jianmo Ni, Jiacheng Li, and Julian McAuley. 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.* 188–197.

[47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. 311–318.*

[48] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* 2227–2237.

[49] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. arXiv:2101.05667. Retrieved from https://arxiv.org/abs/2101.05667

[50] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

[51] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[52] Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 2737–2746.

[53] Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. 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).* 143–155.

[54] Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. Leading conversational search by suggesting useful questions. In *Proceedings of the Web Conference 2020.* 1160–1170.

[55] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for NLG systems. *Computing Surveys* 55, 2 (2022), 26:1–26:39.

[56] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web.* 285–295.

[57] Ivan Sekulić, Mohammad Aliannejadi, and Fabio Crestani. 2021. Towards facet-driven generation of clarifying questions for conversational search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval.* 167–175.

[58] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems.* 172–180.

[59] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval.* 235–244.

[60] Daniel Tunkelang. 2009. *Faceted Search.* Vol. 5, Morgan & Claypool Publishers.

[61] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech and Language* 67 (2021), 101151.

[62] Damir Vandic, Steven Aanen, Flavius Frasincar, and Uzay Kaymak. 2017. Dynamic facet ordering for faceted product search engines. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1004–1016.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems. 6000–6010.*

[64] Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, S. S. Iyengar, Larisa Shwartz, and Genady Ya. Grabarnik. 2019. Online interactive collaborative filtering using multi-armed bandit with dependent arms. *IEEE Transactions on Knowledge and Data Engineering* 31, 8 (2019), 1569–1580.

[65] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *Computing Surveys* 54, 7 (2021), 154:1–154:38.

[66] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining.* 373–381.

[67] Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 2193–2203.

[68] Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 1, 2 (1989), 270–280.

[69] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language–based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 137–145.

[70] Wei Wu and Rui Yan. 2019. Deep chit-chat: Deep learning for chatbots. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1413–1414.

[71] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, and Haiqing Chen. 2020. IART: Intent-aware response ranking with transformers in information-seeking conversation systems. In *Proceedings of the Web Conference 2020.* 2592–2598.

[72] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 974–983.

[73] Shuo Zhang and Krisztian Balog. 2020. Evaluating conversational recommender systems via user simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1512–1520.

[74] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 177–186.

[75] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 83–92.

[76] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do users rate or review? Boost phrase–level sentiment labeling with review–level sentiment classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1027–1030.

[77] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management.* 1411–1420.

[78] Jafar Afzali, Aleksander Mark Drzewiecki, Krisztian Balog, and Shuo Zhang. 2023. UserSimCRS: A User Simulation Toolkit for Evaluating Conversational Recommender Systems. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM'23).* Association for Computing Machinery, 1160–1163. https://doi.org/10.1145/3539597.3573029

Received 31 December 2022; revised 29 May 2023; accepted 5 October 2023