Q&R: A Two-Stage Approach toward Interactive Recommendation

Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, Ed H. Chi
University of Minnesota & Google Inc, Mountain View
christa@cs.umn.edu,(alexbeutel,ruili,sagarj,edchi)@google.com

ABSTRACT

Recommendation systems, prevalent in many applications, aim to surface to users the right content at the right time. Recently, researchers have aspired to develop conversational systems that offer seamless interactions with users, more effectively eliciting user preferences and offering better recommendations.

Taking a step towards this goal, this paper explores the two stages of a single round of conversation with a user: which question to ask the user, and how to use their feedback to respond with a more accurate recommendation. Following these two stages, first, we detail an RNN-based model for generating topics a user might be interested in, and then extend a state-of-the-art RNN-based video recommender to incorporate the user's selected topic. We describe our proposed system *Q&R*, i.e., *Question & Recommendation*, and the surrogate tasks we utilize to bootstrap data for training our models. We evaluate different components of *Q&R* on live traffic in various applications within YouTube: *User Onboarding*, *Homepage Recommendation*, and *Notifications*. Our results demonstrate that our approach improves upon state-of-the-art recommendation models, including RNNs, and makes these applications more useful, such as a $> 1\%$ increase in video notifications opened. Further, our design choices can be useful to practitioners wanting to transition to more conversational recommendation systems.

ACM Reference Format:
Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, Ed H. Chi. 2018. *Q&R*: A Two-Stage Approach toward Interactive Recommendation. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3219894

YouTube Application	Measure of Usefulness	% Improved
User Onboarding	# selected topics	+77.7%
Notifications	notification-opens	+1.23%
Homepage	watch-time	+0.07%

Table 1: *Q&R* leads to a better understanding of user preferences and as a result better user experience in multiple YouTube applications.

1 INTRODUCTION

Recommendation systems play a key role for assisting users to navigate through the vast amount of information available by selecting for them the right item, i.e., product to buy, content to read, video to watch, at the right time [3].

Recently, recommendation researchers and practitioners have aspired to advance the frontier of recommendation by building conversational recommenders in order to create *seamless interactions* with the users. Such systems could better model how real people give recommendations – they try to quickly understand user preferences by asking a few questions under a certain context, and then give a recommendation based on the responses [12]. They should predict the user's potentially evolving, unarticulated interests, while accounting for the fact that users might have a biased view of the world [42]. They aim to help acquire new users by showing interesting content, and retain the existing user base; the two-fold objective of any recommender system to be sustainable.

Recently, this seemingly far-fetched goal has started to seem more tangible [38]. There are mainly two bodies of work towards this goal. On the one hand, a plethora of personal assistants have started to arise in a variety of products across domains, ranging from entertainment or retail bots to health virtual assistants [4, 22, 38] – such systems are powered by recent advances in natural language understanding [44] and focus on conversations, not on recommendation. On the other hand, conversational recommenders, while aiming at recommendation, focus on balancing the explore-exploit tradeoff present in recommender systems [12, 47].

Our work presents a novel view to industrial conversational recommenders. We argue that to transition to truly interactive recommendation systems, we would like components from both of these perspectives. Our system is composed of two parts: a question asking component and an item recommendation one, following the two main stages of a single round of user-system conversation. Further, we need to address how to bootstrap such a system, when there is shortage of data between user and system conversations. For this, we leverage data derived from surrogate tasks we chose from a "traditional" recommender, and bootstrap the components of the system based on these. The result is a novel large-scale learned interactive recommender named *Q&R*, i.e., *Question & Recommendation*. This is the first detailed public description of such a system we know of to-date.

From a modeling perspective, we frame conversational recommendation as factoring out the components of user decisions. This approach lets users give feedback at intermediate states, and generally improves recommendations. Furthermore, we use sequential RNN models to capture the "what next" setting present in conversations, and we intervene by asking users topical questions to allow them to better express their preferences and control their personalized experience. We present our development in YouTube, the

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219894

world's largest video platform [13, 15]; however, our approach is not specific *per se* to video recommendation.

Particularly, our contributions are four-fold:

(1) We present a detailed description of *Q&R*, a **large-scale learned interactive recommendation system**, which asks the users questions about topics and gives item recommendations (Section 3).
(2) To overcome the challenge of the absence of conversational data, we address how to **bootstrap** conversations using large-scale data from a non-conversational recommender, by utilizing surrogate tasks (Section 4).
(3) **RNN-based Two-Factored Recommendation**: We model interactive recommendation in a two-stage setting ("what to ask?", "how to respond?") and propose novel neural-based RNN models for factored recommendation (Section 5).
(4) **Live Traffic Results on YouTube**: We show that *Q&R* can enhance the user experience in multiple applications within YouTube, highlighting the broad impact of our approach (Section 6). Particularly, casual users become 18% more likely to complete the *User Onboarding* experience, and when they do, they select 77.7% more topics. Also, our two-factored video recommendation approach can surface to users more interesting videos to watch, even on top of complex, state-of-the-art RNN recommenders, both in *YouTube Homepage* and in *YouTube Notifications* (Table 1).

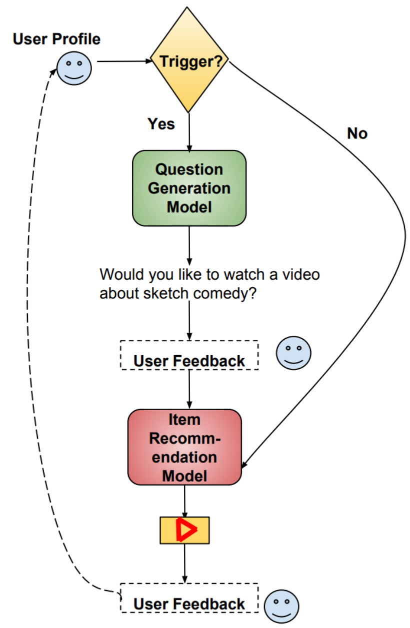

Figure 1: High-level *Q&R* overview.

2 RELATED WORK

Since we study the single round of a conversation between a user and the system, our work can be viewed in the larger context of conversational recommender systems.

The need to present recommendations in a conversational manner [14, 28] has been studied from many perspectives, including interview-based [41], active-learning [39], entropy [47], picture-based [36], explore-exploit [47], critiquing [10], constraint [16], dialog [8], and utility-based strategies [33]. We refer the reader to [18, 22, 38] for a literature review. Here, we compare a few notable works to our system; a comparison overview is found in Table 2.

Many conversational recommendation works have focused on balancing the *trade-off* among exploring the space of user preferences vs exploiting what has been learned thus far [12, 47]; this is a complementary question to our work. Regarding the underlying *model*, existing works use either latent-factor [12] or regression-based models [2]. However, such models have been shown to be outperformed by deeper models when large-scale data is available [13]; thus, we build upon deep Recurrent Neural Networks (RNNs). For the *space of the questions*, most systems ask questions on the end-recommendation items [12, 47]; this is though not possible for domains where the item pool is large and constantly updated, e.g. videos. This is why we ask questions on topics, so as to more effectively propagate feedback among videos sharing the same topic. In terms of the *feedback* elicited, existing systems typically utilize absolute or relative questions or comparisons among two sets of items in a series of questions [31]. Instead, we use a top-N list setting from which the user selects the topics they are interested in [17]. Also, while most existing conversational systems have as *target users* cold-start users [12, 47], we show that our system can improve the experience of existing users as well.

In addition, recent conversational works, although not yet in the context of recommendation, have focused on *natural language understanding* of the user's utterance, and answering knowledge-based questions using sophisticated models (e.g. [26]). However, giving personalized recommendations is a rather different task from answering knowledge questions. In this paper, the interactions are conducted via user clicks; we leave incorporating spoken dialog as future work. Also, most existing conversational recommenders have been demonstrated on a small *scale*; our system is shown in an industrial large-scale setting. With regards to *evaluation*, given the inherent difficulties of ideally having to know how a user would answer every question, most systems have been evaluated using semi-synthetic data [12, 31]; instead, we evaluate *Q&R* on real YouTube users.

3 SYSTEM OVERVIEW

In this paper, we raise the following question: can we improve the experience of casual users (i.e., users who have not visited YouTube in a while, or new users with only a couple of watches in their watch history) by asking them questions on topics they might be interested in?

More generally, would it be possible to bootstrap conversations with users in a large-scale industrial setting?

In answering this question, we face the following challenges:

(1) how to leverage the large **modeling** effort in highly sophisticated **traditional non-conversational recommendation systems**, and still transition to conversational systems?

	Neural Model	Models Sequences	Explicit Context	Low-rank	Conversation Structure	Target Users	Question Space	Scale	Evaluation Data
Critique [10]	✗	✗	✓	✗	*multi-turn*	*cold & warm*	features	small	prototype users
Interactive CF [47]	✗	✗	✗	✓	*multi-turn*	*cold & warm*	items	small	offline recomm.
Abs Pos [12]	✗	✗	✗	✓	*multi-turn*	*cold*	items	small	user study users
Q&R	✓	✓	✓	*approx.* [6]	*single-turn*	*cold & casual*	topics	millions of users	YouTube experiment

Table 2: Relationship with Conversational Recommenders: *Q&R* is the first system to bootstrap conversations based on large-scale user data, demonstrated in live traffic, aimed at new and casual users, and using neural sequence-based models.

(2) **data limitation:** in the absence of conversational data, how can we train our system to ask good questions and adapt well the recommendations after?

To tackle these challenges, we propose *Q&R*, a novel early industrial system for automated question asking and video suggestion after the user's response, demonstrated in the context of YouTube. Particularly, we design *Q&R* to address these challenges as follows:

For the first challenge, we decouple the task of introducing conversations to two sub-tasks: question ranking and item response. In doing so, we can benefit from the vast amount of work in item (in our case video) recommendation. Particularly, we can extend sophisticated models either changing their output space, or the input to incorporate the answer to the question. A by-product of this approach is that not only we make advances to the conversational domain, but we also improve the state-of-the-art in (a) traditional video recommendation, and (b) traditional question (e.g. topic) recommendation.

This approach of decoupling the conversation problem into the question ranking and video response parts aids us with tackling the second challenge as well. Particularly, *Q&R* relies on bootstrapping the models used per task based on data from surrogate tasks. These surrogate tasks can be defined based on the availability of data from already existing tasks of the traditional recommendation system.

We make use of an existing user interface aimed at making the onboarding experience of casual YouTube users better, that we will refer to as "User Onboarding UI" (Figure 2). Although this UI limits us to a single round of conversation, single-round conversation is nevertheless interesting on its own and can give valuable insights into designing a multi-round conversational recommender.

3.1 Design Goals

While designing *Q&R*, our main goal is the improvement of user experience after the inclusion of our system. We hypothesize that to achieve this, we need improved accuracy in:

(1) *question ranking* quality, and
(2) *response relevance* after user's feedback

and we test this hypothesis with our experiments.

Other aspects we consider for the system design are: *scalability*, to scale well with a large pool of items for recommendation, *temporal* patterns based on user sequence data and *freshness* of results and conversations to keep track of newly generated content.

3.2 Main Components

Q&R consists of the following components, also shown in Figure 1:

(1) **Question Generation** (Section 5.2): a deep sequential network that predicts which topics will interest the user.
(2) **Item Recommendation** (or else **Response**) (Section 5.3): a deep sequential network that predicts given the selected topics, which videos should be recommended.

3.3 The Life of An Interaction

A user arrives in YouTube. This user is characterized by the user profile, which is a set of features that describe the user and their history of interactions with the system.

Based on a *triggering* mechanism, either the *question generation* or the *item (video) recommendation* module is used. The *triggering* mechanism can be as simple as a random mechanism that decides whether the user should be led to a conversational experience, or can be more sophisticated, i.e., using criteria capturing the user's state, or even be user-initiated. When the *triggering* decides *No*, a traditional item recommender is used. Else, a two-stage conversational approach is invoked.

If the triggering mechanism decides *yes* (e.g. a new or casual user arrives at YouTube), the user is directed to the conversational experience, which is materialized via a "User Onboarding" UI. This UI, which is prevalent for new users in recommendation systems (e.g. Facebook, Pinterest, Flipboard), presents a personalized list of topics, the *question generation* module has chosen, from which the user is prompted to choose as many topics as they like, with the understanding that they will be used to improve their experience in the content feed. An illustration is shown in Figure 2.

Once the question (top-1) or list of questions (top-k) has been asked, the user provides feedback. In this work we focus on positive-only type of feedback, i.e., which of the questions/topics in the User Onboarding did the user select? Given the user feedback, a *response* module is used to adapt the user experience. The response module incorporates the user feedback with the goal of improving video recommendations. Ideally, for every one of the topics selected in the Onboarding UI, the user should be able to find at least one relevant item on that topic.

4 PROPOSED SURROGATE TASKS

We now turn our focus to one of the central themes of this paper: Can we leverage the user data and other signals available from traditional interfaces, to design the new paradigm of conversational recommenders? We answer this question positively by framing proxy tasks, based on which we train predictive models.

For the purposes of this work, we consider a single round of a conversation. We use as inspiration how an actual effective conversation between two people would go: Person A asks a question, or gives a prompt, and then person B replies to this question. Then to complete a full round of conversation, person A is supposed to (1) understand what has been said by person B, relevant to what A

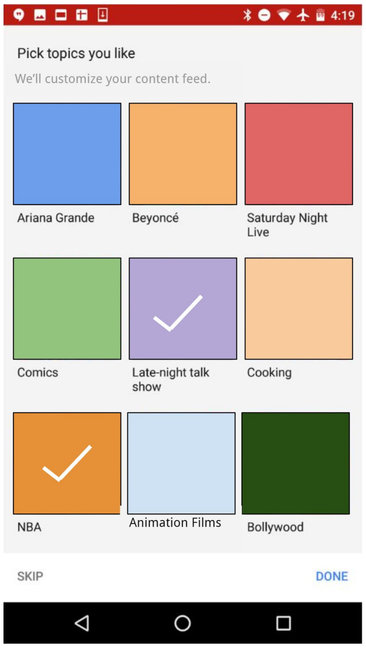

Figure 2: User Onboarding UI.

asked, and (2) adapt their mental model of the conversation state, based on person B's response.

Similarly, in one round of a conversation between the recommendation system (A) and the user (B), the conversational recommender is supposed to be designed for two tasks: (1) decide what to ask to the user, (2) decide how to adapt the response and change its model about the user, according to the user's provided feedback.

In what follows, we focus on formalizing these tasks and the precise training procedure for each; in Section 5 we present the machine learning models we build for these tasks.

4.1 Proxy Task for: What To Ask

For the surrogate task used for the question ranking module, ideally we need data of good questions which were asked by a triggered conversational system, given a user profile. Instead, we introduce the surrogate task: *what is the topic of the next video a user would want to watch*?

For this, we consider the watches of sampled users, and split these into two parts: the sequence of watched videos up until t, and the watched video for $t + 1$. We then use the watch sequence data to predict the most relevant topic ID associated with the next video to be watched (noting that the future video ID is not given as input). This setup can capture a user's topical interest in a just-in-time recommendation setup.

We chose the task of predicting the clicked video's topic as opposed to predicting other user signals such as user's search query/ comment(s) as we wanted to: (i) capture unarticulated user interest, which can come directly from getting the topics of the future watched videos, and (ii) to ask questions which cover a large interest space (instead of focusing e.g. on the most popular queries/ words of comments).

In essence, for tackling the what-to-ask task, we reduced the problem to the one of building *better user profiles* [9], i.e., predicting the sequential future in terms of what topic a user will be interested in. This serves a double purpose: first, show the user that the recommender understands their preferences (establish user trust), and second, the more accurate the personalized topic recommendations in the what-to-ask module are, the more the chances that the user will click on these topics and consequently on videos in the homepage – depending of course, on whether the response module can find relevant interesting videos for these topics.

4.2 Proxy Task for: How to respond

Given that the user has been shown a question and has responded to it, how should the system incorporate the feedback and give a recommendation? Since we do not have access to data from successful one-round conversations, we instead introduce this proxy task: *given the most relevant topic of the to-be-watched video, what video will the user be most interested in?* For this, we divide the watch history data of sampled users to two parts: (1) the sequence of watched videos up until t, along with the topic ID of the watched video in t+1, and (2) the watched video for t+1, and we use part (1) to predict part (2). Here, we use the topic ID of the future-watched video to play the role of the user's provided feedback, i.e., given that the user wants to watch e.g. a *late night show* video, what video should we recommend?

5 MODELING

The core of *Q&R* is a statistical model for generating potential questions for the user (5.2), and a statistical model that, given the user's answer and other information, generates the adapted item recommendations (5.3).

5.1 Using Sequential Neural Models

For both models, we adopt a sequential approach, since we wish to predict the sequential future given the past. In particular, we build on recent advances in recommendation systems, which pose the problem of top-N recommendation as a *sequence-to-one learning* problem [6, 45]: Given the sequence of events (e.g. watches), what is the next video a user will watch? For the next timestep of the sequence, the problem can be framed as a multi-class classification problem, where each video is a different class. This is different from the rating prediction view (how many stars will a user give to the video?), classification view (will a user click on this video?), or the ranking view (rank all videos for a user) which have been largely used to model recommendation [3].

Intuitively, the main building block of a sequential recommender is a Long Short Term Memory (LSTM) [20] unit or a Gated Recurrent Unit (GRU) [11], due to their ability to capture long and short term dependencies, which might be present in user behavioral patterns. Typically, a recurrent unit maps the input, which can be a sequence of features (or feature embeddings), to a latent vector that captures the state. This is usually followed by a softmax unit that maps the latent state to the class probabilities.

The benefit of adopting a sequential learning approach is that user preference drifts are better captured, and temporal patterns are learned [21, 24, 45]. This is crucial for delivering the right content at

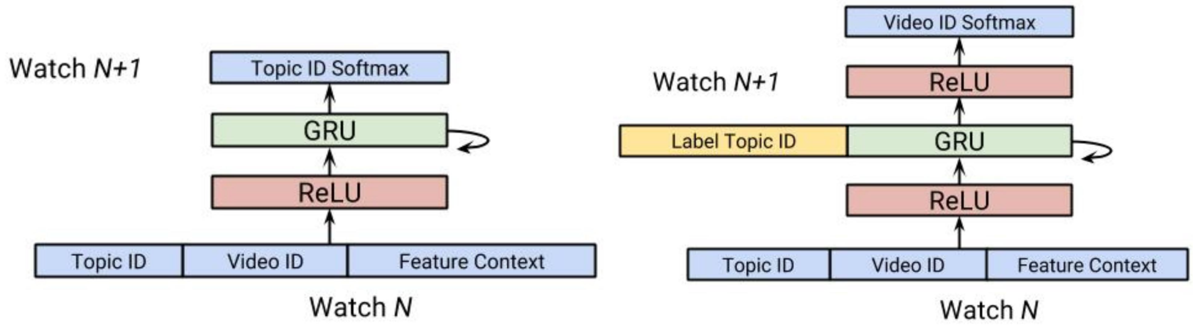

Figure 3: *Left:* Topic Prediction (Question Ranking) Model. *Right:* Post-Fusion Approach for Response Model.

the right time. This modeling view aligns wells with our formulation of predicting the topics a user will like in in the future, without her having even expressed any interest in them in the past. It also offers a good test-bed for future work on multi-round conversations, as the recurrent view can remember long-term dependencies.

5.2 Proposed Question Ranking Model

Since we are predicting the topic question q, conditioned on a sequence of watch events $\mathbf{e}$, this problem is a natural fit for sequence-to-one learning. The sequential unit we use is a GRU due to its good empirical performance [11]. The input is the watch events of the user history up to T time-steps back $\{e_1, \ldots, e_T\}$, and the output is the conditional probability distribution of the topic a user will be interested in at time $T + 1$[1], given the input:

$$P(q|e_1, \ldots, e_T) \tag{1}$$

As shown in Figure 3(*Left*), the sequence of watch events are fed in, such that the GRU's hidden state encodes a vector representation of the entire watch history. Then, given this hidden state, a softmax output is computed and gives the probability distribution over the topic corpus in the next timestep.

Training. Given a large corpus of users' watch histories, the objective is to maximize the log probability of observed topic events, given the respective past watches:

$$\sum_{(e,q)} \log P(q_{T+1}|e_1, \ldots, e_T) \tag{2}$$

We train against this objective, which is categorical cross entropy loss, using stochastic gradient descent. Training is run in a distributed fashion using the TensorFlow library [1].

Inference. At inference time we feed in the user sequence and use the softmax output to get a probability distribution over the topic corpus for the final timestep. This distribution is used by showing the topic with the largest probability (or top-k topics with the largest k probabilities).

Using this model for question ranking, notice that the questions are on topics instead of specific items, in contrast to prior work. This allows us to utilize user feedback on topics to propagate information to the varying and much larger space of videos.

[1] A natural future direction is to pose this in a sequence-to-sequence framework, predicting topics for the next k steps, i.e., for $T + 1$ until $T + k$

5.2.1 Relationship to Literature on Topic Recommendation. A vast literature exists on personalized topic prediction or more generally on tag [7, 27, 29]/ topic [34] /query [5] recommendation. Most of these works have focused on discovering latent topics, typically using topic models [25, 37] or graph-based algorithms [43, 46], whereas in our case topics are not latent. When topics are explicit, most works rely on collaborative filtering [32, 40, 48] or information retrieval solutions [35]. To the best of our knowledge, although sequential approaches have been successful for conversational bots [23, 44], and have been applied in the context of session-based recommendation [19] or item recommendation [45], this is the first time to be applied on topic recommendation.

5.3 Proposed Video Response Model

Imagine the user has selected the topic q they are interested in. To satisfy the user's topical interest, the video recommendations need to be adapted to reflect this interest. Mathematically, we want to maximize the probability of $P(r|e_1, ..., e_T, \text{topic}{=}q)$, capturing the goal of finding, given the user watch history $\mathbf{e}$ and the clicked topic q, a good response r (a.k.a the videos that the user tends to watch next). We discuss two alternative approaches, Restricted Output and Post Fusion, that we empirically compare in Section 6.1.

Restricted Output. The simplest approach is to use an already trained video GRU-based recommendation model [6], but during inference restrict the softmax output vocabulary to be only on videos associated with this topic, i.e., with the understanding that topic is a given feature associated with each video. The trained model has the same architecture as the topic RNN shown in Figure 3(*Left*), except the softmax is over the vocabulary of videos which are about a certain topic q. In fact, this model architecture is a simpler version of the model proposed in [6].

Post Fusion. An alternative is to train a sequential model to predict the last video of the watched sequence, given the past watch events and the topic of the to-be-predicted video. The model is again GRU-based. The input is exactly the same as in the question ranking task, with the only difference that the GRU hidden state output is concatenated (fused) with the embedding of the topic ID associated with the to-be-predicted video. We give the topic information as post-fusion, i.e., after the GRU, as opposed to pre-fusion, i.e., before the GRU, to maximize the influence of the topic on the softmax function during back-propagation. The concatenated GRU output with the topic embedding is passed through a ReLU unit (we found that it helped empirically) and then, to a softmax over the videos. Here, the output vocabulary is not restricted. The model is shown in Figure 3(*Right*).

Training. Given a large corpus of users' watch histories, the objective is to maximize the log probability of observed videos, given the respective topics associated with these videos and the sequence of past watches:

$$\sum_{((\mathbf{e},q),r)} \log P(r_{T+1}|e_1,\ldots,e_T,q) \tag{3}$$

Inference. The information of the topic associated with the future video watch is known only during training. During inference, we pass the actual topics selected by the user in the conversation. For each topic, we use the trained model to infer the top-K video recommendations. Then, a post-process ranking method is used to blend the various top-K recommendations from the different topics.

5.4 Proposed Two-Factored Approach

Combining the two models described in Sections 5.2 and 5.3, we can model the probability of a user watching a video in $T+1$ as:

$$P(q_{T+1}|e_1,\ldots,e_T)\cdot P(r_{T+1}|q_{T+1},e_1,\ldots,e_T) \tag{4}$$

In other words, we can factor item recommendation in two models: one model to predict the personalized topics for the user profile, and a second model where item recommendations are given *pertinent to these inferred topics* [30, 49].

Our hypothesis is that such an approach could make the problem of item recommendation easier, by correctly predicting the topic a user will like, and constraining the video recommendation space to videos related to this topic.

6 EVALUATION AND RESULTS

We perform three sets of experiments. The first is on a dataset of YouTube user sequences, and we compare different models for the separate tasks of question asking and video response (Section 6.1). In the second set of experiments, we explore the relative improvements in video recommendation when we evaluate our two-factored approach on real YouTube users in two applications: YouTube Homepage and YouTube Notifications (Section 6.2). With the goal of evaluating the topic ranking component of *Q&R*, we perform the third experiment on the User Onboarding UI which asks real YouTube users to select topics they are interested in to personalize their YouTube experience (Section 6.3).

6.1 Offline Comparative Analysis

6.1.1 Evaluation Setup. **Data.** Regarding the data used to train our models, we followed the approach described in Section 4 for the proxy tasks. We sampled YouTube user watch sequences, and we divided each sequence into two parts: (1) the watch sequence of a user up until the previous to last step, and (2) the video ID and topic ID of the user's last watch event. The data contains watch sequences of hundreds of millions of users. Watches are restricted to a large corpus of popular videos (millions) and all users have at least a number of watches (tens) in their sequence. The users are split into training, validation and test sets, with both validation and test sets having tens of millions of users. The sequence is given by a list of watched videos and the timestamp of each watch.

Every watched video is associated with the following features: video ID, ID of the most relevant topic associated with the video, ID of the video's creator (channel), features showing the device type on which the video was watched, timestamp information capturing the temporal component, a cross-product transformation of some pairs of features, and others [6]. All features except for video ID and topic ID were shown as contextual input features in Figure 3. Each categorical feature has its own vocabulary (unique ID space), and the ID of the value is mapped to a dense embedding. Out-of-vocabulary values are mapped to the zero embedding. The embeddings of the categorical features are concatenated with the numerical features and passed through the GRU unit to capture the entire watch history.

Parameters. We used the validation set to tune our model parameters. After tuning, we set the embedding size, hidden unit dimension and softmax embedding dimension to 256. The size of the vocabularies for videos and topics are in the order of millions and hundred thousands respectively. The softmax layer, over the video/ topic corpus for video/ topic recommendation, is trained using sampled softmax with tens of thousands negative samples per batch [6].

Implementation. Our models were built with TensorFlow [1], and learned with stochastic gradient descent using categorical cross-entropy until convergence.

Training/Test Split. We used as training data the watch sequences from the period of seven consecutive days, and evaluated against the held-out data of the eighth day. This setup ensures that training and evaluation data are disjoint, and that we indeed predict the sequential future [13]. When evaluating against the eighth day data, we feed as input the sampled watch sequences of these data up until the previous to last timestep, and infer the topic/video watched in the final timestep (similarly to the training procedure).

Metric. We use Mean Average Precision (MAP) @ top k as our offline metric reported in the evaluation data. This is because we want to capture the top recommendation quality of our question ranking and video response models. The value of k is typically small, i.e, 10-20, as users rarely scroll down after 10-20 recommendations.

MAP@k is defined as the mean over the Average Precisions (AP@k) of the users, where for a single user AP@k is given by:

$$AP@k = \sum_{r=1}^{k} \frac{rel(r)}{\min(k, \text{num. of relevant items})} \tag{5}$$

where $rel(r)$ for the item at position r is 1 if the user has watched the video/topic, and 0 otherwise. The ideal value for MAP@k is 1.

6.1.2 Results on Topic Prediction. First, we evaluate the predictive quality of the question (topic) ranking component. We report MAP@20 results, noting that MAP@1 results follow similar trends.

In Figure 4 we compare the MAP@k of our *Q&R* Topic RNN model (Figure 3, *Left*) to three baselines:

- *Random:* which ranks topics from the set $q : Q$ randomly.
- *Most Popular*: which ranks topics according to the aggregated number of watches of the videos associated with the topic.
- *Multiclass-BOW*: which ranks the topics using a Bag-of-Words style model over the event sequence.

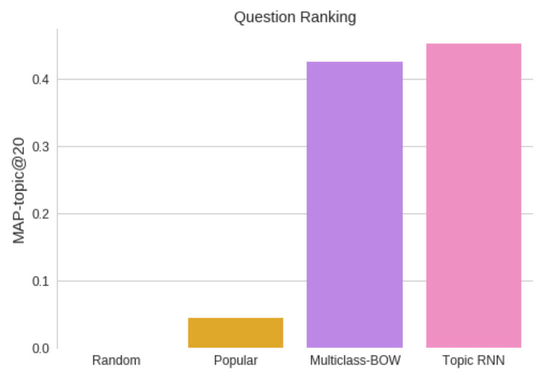

Figure 4: MAP-topic@20 question ranking results.

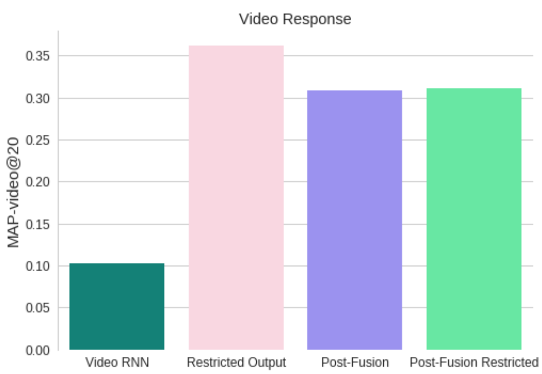

Figure 5: MAP-video@20 video response results.

The *Random* baseline helps us understand the complexity of the task of topic prediction. We can see that *Random* essentially achieves a MAP score close to 0. *Q&R Topic RNN* is significantly better than the *Most Popular* greedy state-less baseline. This shows that conditioning on the input watch sequence of the user is effective for accurately ranking questions, and that keeping track of the *state* of the conversation is important for building a good topic prediction model. In addition, we observe that our *Q&R Topic RNN* outperforms the *Multiclass-BOW* by 8.07%; this demonstrates that using an RNN unit to capture the sequential nature of the data allows the model to learn more sophisticated representations compared to using the bag of word features. This experiment validates our modeling of topic prediction as a sequential problem.

6.1.3 Video Response Prediction Results. Next, we evaluate the predictive quality of the video response component. We compare our two proposed approaches: *Restricted Output*, and *Post-Fusion* with an RNN video recommendation model which is not conditioned on the topic used, which we refer to as *Video RNN*. The difference between *Video RNN* and *Restricted Output* is that during inference, the latter's softmax output is restricted to be over the corpus of all videos which are associated with the most relevant topic of the video to be predicted, whereas in *Video RNN* such a restriction does not apply.

Figure 5 presents the MAP@20 of these video response models. By comparing any of the methods conditioned on the topic of the to-be-predicted video (e.g. *Restricted Output, Post-Fusion*) with *Video RNN*, we can see that this conditioning on the topic (used as a post-fusion feature, or as a restriction on the output vocabulary) can lead to more than 3 times better recommendation accuracy.

Also, comparing our two discussed approaches for video response (Section 5.3), i.e., *Restricted Output* with *Post-Fusion*, we find that *Restricted Output*, with the considerably smaller output space and fewer parameters to be learned, can achieve better predictive quality. This is also the case even if we compare a hybrid approach of *Post-Fusion Restricted* with *Restricted Output*: *Restricted Output* continues to be better. We hypothesize that this behavior is due to the fact that the restrict part is limited by the ability of post-fusion.

6.2 Evaluating *Q&R* on YouTube

6.2.1 Evaluation Setup. The goal of this experiment is to answer the question: Does our approach of factoring video recommendation to two-fold recommendation models, (i.e., (1): user history → topic recommendation, and (2): user history & top-K topics from (1) → video recommendation), lead to a better user experience compared to a direct video recommendation model?

Models. To accomplish this, we use for model (1) our *Q&R* Topic RNN model, and for (2) the *Restricted Output* approach thanks to its simplicity to be used, as it requires small changes on top of a maintained production video recommender. We compare to the production system, which already includes a production RNN-based baseline for (one-fold) video recommendation. We follow [13] and use a two-step approach of candidate generation and ranking, where the ranker is a deep feed-forward neural network which given the generated candidates by many nominator models, ranks the top-k candidates.

Live Traffic Testbeds. We opted to evaluate our novel two-factored approach for video recommendation (what topic, and what video given the topic) on two separate modules of YouTube: (1) YouTube Homepage "topic shelf" and (2) YouTube Notifications, and we present our results in each one, in what follows.

Setup. To ensure that the topical questions and video recommendations are up-to-date with the users' temporal preferences, the pool of items for recommendation is updated frequently to include fresh content based on new user-item interactions. The topic vocabulary is updated by collecting all the topics associated with each item in the item vocabulary. For freshness, both the question generation and response models are periodically re-trained to incorporate new user-system interactions, using also the freshly updated vocabularies.

We run our live experiment over a small slice of the entire user traffic in YouTube during the period of two weeks. To serve our models in live traffic, we add our *Q&R* model (*Topic RNN* followed by *Restricted Output*) in the pool of nominator models; then, the ranker model interlaces video recommendations from the various nominators for the content feed. We then measure which of the nomination algorithms can find the video recommendations that will better capture the user's interests.

Metric. Here, we want to measure whether our two-factored approach can better understand user preferences, and can thus find

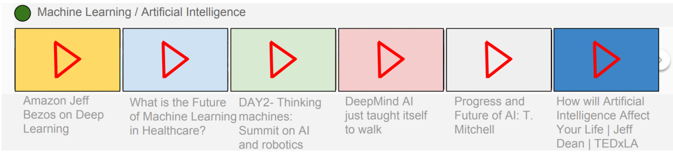

Figure 6: Topic Shelf related to the topic 'Machine Learning'.

Measure of Usefulness	% Improved
# selected topics	+77.7%
completion probability	+18%
watch-time	+4%

Table 3: Live metric % improvement of *Q&R* Topic RNN vs. topic ranking baseline in User Onboarding.

more interesting videos that users would like to watch. As proxy metrics for utility to users, we report the time spent watching videos (*watch-time*) when testing recommendations on the homepage, and when testing recommendations in notifications we measure the number of users who open the notifications (*notification-opens*).

6.2.2 Results on YouTube Homepage. The first UI on which we evaluate our methods is the YouTube Homepage "Topic Shelf". As illustrated in Figure 6 this UI consists of topical shelves each of which is associated with a topic; and the topic can be as general as *TV show* or *funny videos*, or as specific as *Late Night TV Shows*, or *mexican cooking recipes*. In mobile interfaces the UI is slightly different, with the topic-triggered video recommendations being given inline (not as separate rows) with the other videos nominated by one-fold video recommendation models. The task is to select which topic shelves to show (*question ranking*), and which videos to fill these topics shelves with (*video response*).

After including our two-factored approach in the pool of nominator models for video recommendations, we observed that on average our models result in a 0.07% improvement in the time spent watching videos, compared to the production baseline – a highly optimized baseline, including RNNs, that is hard to compete with. This validates the usefulness of our two-fold approach. We hypothesize that the improvement happens because the topic recommender can find interesting topics for the user, thus making the video recommender's job easier given the inferred topics.

6.2.3 Results on YouTube Notifications. The second experiment that we conduct on live traffic is for YouTube Notifications. In this scenario, new video recommendations are sent daily to users who have opted-in to receive their video recommendations via notification to their devices. At most one video recommendation is sent to a user. Here, we followed the same approach as [13] to compute video recommendations in two steps, which include candidate generation and ranking, and we added our *Q&R* model to the pool of models nominating videos for ranking.

After adding our two-factored approach in the pool of nominators in our experiment, we observed that on average our models result in a 1.23% improvement in terms of the number of users who open the recommended video notifications, compared to the production baseline. Here, we again argue that the production baseline is a very strong baseline to compete with, which already includes an RNN-based (one-fold) video recommendation model. It again verifies the value of our proposed two-factored approach, and the effectiveness of modeling the two steps with RNN models.

6.3 Results on User Onboarding

In our final set of experiments, we focus on evaluating the *Q&R* topic ranking component in the context of creating a more enjoyable experience for a casual user via a list-format of a conversational experience. Particularly, the users are directed to the User Onboarding (Figure 2) which allows them to select a number of topics from a list of 100 personalized topics or completely skip the Onboarding UI. Out of the entire user traffic directed there, we run our live experiment over a small slice for the period of three days.

Our goal in this experiment is to compare only the question generation component of *Q&R*; a response mechanism resembling *Restricted Output* was already in place.

We compared *Q&R Topic RNN* with the existing production baseline that produces the personalized list of topics. The baseline is a *Naive Bayes* method that takes as input user features including watch histories to produce top-K topic recommendations. To do this comparison, we performed an A/B test, where in A (control) we used the Naive Bayes and in B (treatment) we employed our recurrent solution.

In Table 3 we report by how much metrics capturing user satisfaction in (and after) the Onboarding UI improve when using *Topic RNN* for ranking the topics in the UI. We observe a 4% improvement on average in the amount of time spent watching videos among users in the Onboarding UI who had the RNN topic recommendations treatment, compared to those in the control set of the Naive Bayes baseline. We further find that on average with our *Q&R* Topic RNN, the number of topics selected goes up by 77.7 % compared to the one achieved by the baseline. Also, we find that the probability of completing the Onboarding UI goes up by 18 %. The results from this experiment validate the idea that our RNN-based topic recommender can find accurate personalized topics; thus making the Onboarding UI more useful and improving users' experience after interacting with this UI.

Overall, we can see that the *Q&R Topic RNN* method significantly outperforms the *Naive Bayes* baseline. This shows that despite the small length of the user watch sequence (given that triggered users are casual ones), the hidden state vector learned in *Topic RNN*, is more informative compared to the hand-crafted input features, for creating an accurate personalized topic list for the user. This gives real-world insight that our approach makes it easier for users to express their preferences, and puts them in better control of their personalized experience.

7 CONCLUSIONS

To the best of our knowledge, this is the first work on learned interactive recommendation (i.e., asking questions and giving recommendations) demonstrated in a large-scale industrial setting.

Our work brings attention to the often overlooked problem of bootstrapping conversations based on interactions from a traditional system. We believe that our discussion on the different design choices we made and the proxy tasks we utilized could help practitioners make faster progress on transitioning to more conversational systems.

In building *Q&R*, we set out to improve the user experience of casual users in YouTube. Users become 18% more likely to complete the User Onboarding experience, and when they do, the numbers of topics they select goes up by 77.7%.

In the process, we provide a novel neural-based recommendation approach, which factorizes video recommendation to a two-fold problem: user history-to-topic, and topic& user history-to-video. We demonstrate the value of our approach for both the YouTube Homepage and YouTube Notifications.

Finally, having shed light on a single round of conversation, the area of research in industrial conversational recommendation systems seems to be wide-open for exploration, with incorporating multi-turn conversations and multiple types of data sources, as well as developing models for deciding when to trigger a conversational experience, being exciting topics to be explored in the future.

ACKNOWLEDGMENTS

We thank very much Vince Gatto and Paul Covington as without them, this work would not have been possible. Also, we thank the reviewers for their insightful comments, Francois Belletti and Minmin Chen for useful discussions, and the entire SIR team for feedback on earlier versions of this work.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Deepak Agarwal and Bee-Chung Chen. 2009. Regression-based latent factor models. In *KDD*. 19–28.

[3] Xavier Amatriain. 2012. Building industrial-scale real-world recommender systems. In *RecSys*. 7–8.

[4] George Anders. 2017. Alexa, Understand Me. https://www.technologyreview.com/s/608571/alexa-understand-me/. Accessed: 2017-10-25.

[5] Ricardo A Baeza-Yates, Carlos A Hurtado, Marcelo Mendoza, et al. 2004. Query Recommendation Using Query Logs in Search Engines. In *EDBT workshops*, Vol. 3268. 588–596.

[6] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *WSDM*. 46–54.

[7] Ágnes Bogárdi-Mészöly, András Rövid, Hiroshi Ishikawa, Shohei Yokoyama, and Zoltán Vámossy. 2013. Tag and topic recommendation systems. *Acta Polytechnica Hungarica* 10, 6 (2013), 171–191.

[8] Derek G Bridge. 2002. Towards Conversational Recommender Systems: A Dialogue Grammar Approach. In *ECCBR Workshops*. 9–22.

[9] Cheng Cao, Hancheng Ge, Haokai Lu, Xia Hu, and James Caverlee. 2017. What Are You Known For?: Learning User Topical Profiles with Implicit and Explicit Footprints. In *SIGIR*. 743–752.

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

[11] 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. *arXiv preprint arXiv:1406.1078* (2014).

[12] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *KDD*. 815–824.

[13] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.

[14] Ingemar J Cox, Matt L Miller, Thomas P Minka, Thomas V Papathomas, and Peter N Yianilos. 2000. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *IEEE transactions on image processing* 9, 1 (2000), 20–37.

[15] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *RecSys*. ACM, 293–296.

[16] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. 2011. Developing constraint-based recommenders. In *Recommender Systems Handbook*. Springer, 187–215.

[17] Mark P Graus and Martijn C Willemsen. 2015. Improving the user experience during cold start through choice-based preference elicitation. In *RecSys*. 273–276.

[18] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016), 9–27.

[19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *WSDM*. 515–524.

[22] Michael Jugovac and Dietmar Jannach. 2017. Interacting with recommenders – overview and research directions. *TiiS* 7, 3, 10.

[23] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, László Lukács, Marina Ganea, Peter Young, et al. 2016. Smart reply: Automated response suggestion for email. In *KDD*. 955–964.

[24] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[25] Ralf Krestel and Peter Fankhauser. 2009. Tag recommendation using probabilistic topic models. *ECML PKDD Discovery Challenge* 2009 (2009), 131.

[26] Huayu Li, Martin Renqiang Min, Yong Ge, and Asim Kadav. 2017. A Context-aware Attention Network for Interactive Question Answering. In *KDD*. 927–935.

[27] Huizhi Liang, Yue Xu, Dian Tjondronegoro, and Peter Christen. 2012. Time-aware topic recommendation based on micro-blogs. In *CIKM*. 1657–1661.

[28] Greg Linden, Steve Hanks, and Neal Lesh. 1997. Interactive assessment of user preference models: The automated travel assistant. In *User Modeling*. Springer, 67–78.

[29] Dong Liu, Xian-Sheng Hua, Linjun Yang, Meng Wang, and Hong-Jiang Zhang. 2009. Tag ranking. In *WWW*. 351–360.

[30] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*. 31–40.

[31] Benedikt Loepp, Tim Hussein, and Jüergen Ziegler. 2014. Choice-based preference elicitation for collaborative filtering recommender systems. In *CHI*. 3085–3094.

[32] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-Based Collaborative Filtering for News Topic Recommendation. In *AAAI*. 217–223.

[33] Tariq Mahmood and Francesco Ricci. 2009. Improving recommender systems with adaptive conversational strategies. In *HT*. 73–82.

[34] Anirban Majumder and Nisheeth Shrivastava. 2013. Know your personalization: learning topic level personalization in online services. In *WWW*. 873–884.

[35] Nicolaas Matthijs and Filip Radlinski. 2011. Personalizing web search using long term browsing history. In *WSDM*. 25–34.

[36] Julia Neidhardt, Rainer Schuster, Leonhard Seyfang, and Hannes Werthner. 2014. Eliciting the users' unknown preferences. In *RecSys*. 309–312.

[37] Sanjay Purushotham, Yan Liu, and C-C Jay Kuo. 2012. Collaborative topic regression with social matrix factorization for recommendation systems. *arXiv preprint arXiv:1206.4684* (2012).

[38] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *CHIIR*. 117–126.

[39] Neil Rubens, Dain Kaplan, and Masashi Sugiyama. 2015. Active Learning in Recommender Systems. *Recommender Systems Handbook*, 809–846.

[40] Börkur Sigurbjörnsson and Roelof Van Zwol. 2008. Flickr tag recommendation based on collective knowledge. In *WWW*. 327–336.

[41] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *WSDM*. 445–454.

[42] Yu Sun, Nicholas Jing Yuan, Yingzi Wang, Xing Xie, Kieran McDonald, and Rui Zhang. 2016. Contextual intent tracking for personal assistants. In *KDD*. 273–282.

[43] George Toderici, Hrishikesh Aradhye, Marius Pasca, Luciano Sbaiz, and Jay Yagnik. 2010. Finding meaning on youtube: Tag recommendation and category discovery. In *CVPR*. 3447–3454.

[44] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* (2015).

[45] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. 495–503.

[46] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*. 283–292.

[47] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *CIKM*. 1411–1420.

[48] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H Chi. 2015. Improving user topic interest profiles by behavior factorization. In *WWW*. 1406–1416.

[49] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW*. 22–32.