Step-wise Recommendation for Complex Task Support

Elnaz Nouri, Robert Sim, Adam Fourney, and Ryen W. White
Microsoft Research
Redmond, WA
{elnouri,rsim,adamfo,ryenw}@microsoft.com

ABSTRACT

Digital assistants help people perform simple tasks, including scheduling, home automation, information look up, and question answering. Current assistants offer less support for accomplishing more complex tasks comprising multiple steps. These tasks sometimes require the ability to leverage the capabilities of multiple devices. Ideal smart assistants for such complex scenarios would track task progress and recommend useful and appropriate information at every step of the procedure. To this end, we introduce the novel notion of *step-wise recommendation* as a means of automatically providing guidance relevant to the current *step* in a complex task. We employ a common real-life scenario for this purpose: recipe preparation. We demonstrate how a smart assistant can be developed to offer support for complex tasks enabled by multi-modal inputs and outputs through multiple devices (i.e., smart speakers, tablets, or other smart devices available in kitchens). We develop step-wise recommendation models for this scenario and analyze their efficacy for: (1) different prediction tasks (e.g., resources, devices), and (2) different contextual information used to make the prediction (e.g., completed steps, current step, and importantly, future steps). Our recommendation model achieves a prediction accuracy of 83-96%, depending on the prediction task and context used. The findings have implications for the design of intelligent systems to help people accomplish complex tasks.

CCS CONCEPTS

• **Human-centered computing → Interaction paradigms**; • **Information systems → Recommender systems**; • **Computing methodologies → Machine learning**.

KEYWORDS

Step-wise recommendation; Multi-device experiences; Complex tasks; Intelligent assistants; Conversational systems

ACM Reference Format:
Elnaz Nouri, Robert Sim, Adam Fourney, and Ryen W. White. 2020. Step-wise Recommendation for Complex Task Support. In *2020 Conference on Human Information Interaction and Retrieval (CHIIR '20), March 14–18, 2020, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3343413.3377964

CHIIR '20, March 14–18, 2020, Vancouver, BC, Canada
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6892-6/20/03...$15.00
https://doi.org/10.1145/3343413.3377964

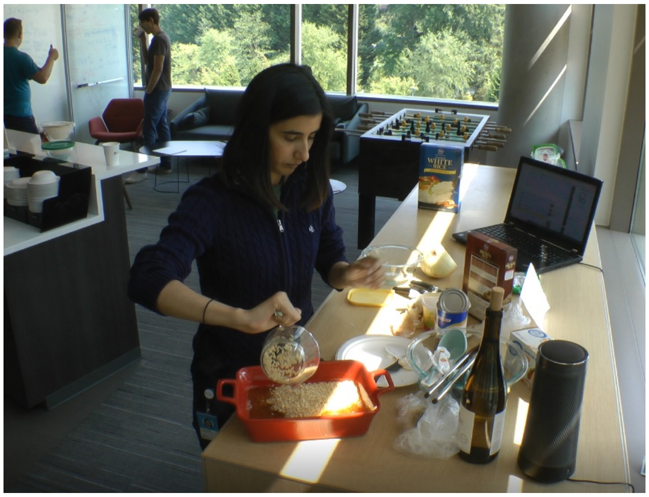


Figure 1: Example of a user cooking a recipe with assistance from a multi-device cooking application, *AskChef*. The application provides a recommendation for every step and leverages the capabilities and modalities of the devices available at cooking time (in this case, a smart speaker (lower right of the figure) and a laptop (middle right of the figure).

1 INTRODUCTION

There has been a steady growth in the popularity of smart digital assistance in recent years [1, 37]. This trend is forecast to continue, with a predicted compound annual growth rate exceeding 50% by 2020 [50, 67]. Digital assistants (DAs) such as Amazon Alexa, Google Assistant, and Microsoft Cortana can already help their users accomplish simple tasks such as playing music, checking weather, and controlling smart devices in their environment (e.g., appliances, lighting, thermostats) [31]. It is natural to expect that the capabilities of DAs will continue to grow, e.g., to span multiple devices that users already own and have at their disposal, such as smartphones, smart wearable devices, tablets, and desktop (or laptop) computers. An example of this type of multi-device assistance with two devices (a laptop and a smart speaker) is shown in Figure 1. Each device has unique strengths in aspects such as display, compute, portability, sensing, communications, and input. Greater availability of cloud services, coupled with the large number of digital devices available in people's daily environment, create unique opportunities for DAs to support multi-device scenarios and provide guidance across a wider range of complex tasks that people perform regularly (e.g., cooking a new recipe, home or auto repair, or furniture assembly) [69]. Assisting people with complex tasks, often involving multiple steps, requires a sound understanding of the task, and the current status of the user, as well as knowledge of the portfolio of devices available to the user at the current time [32].

Towards this goal, in this research we introduce the novel notion of *step-wise recommendation*, a means to provide users with support for the completion of complex tasks based on their status in the task and different sources of contextual information. Step-wise recommendation enables DAs to leverage the knowledge of all steps in a task, including future steps (if available), to help guide users as they make progress.

For decades, researchers have studied the recommendation of interesting items based on the ratings of others [23], user profiles [48], and the combination of these sources [39]. Prior work has also used implicit signals from task contexts and user histories for recommendations [2, 15], including task dialogs [71] and session data [56]. However, there has been less focus on support for sequential tasks and on using the content of past, current, and future steps (often known in advance for multi-step tasks such as cooking) to provide appropriate recommendations. Our research on step-wise recommendation is substantiated based on a working prototype of a multi-device experience (MDX) developed to help people prepare recipes [69] (Figure 1). Recipe preparation (our chosen domain) is an excellent example of a complex multi-step task, in which different formats of assistance (e.g., voice and visual assistance deployed from smart speakers and screens) are helpful to users, who need to allocate attention to following recipe instructions, as well as execution with their hands and ingredients in the physical world.

Task support for recipe preparation, and complex tasks in general, requires not only knowledge of the steps in the task and relevant digital resources, but also physical objects such as available companion devices and their capabilities (e.g., recommending a user turn their attention to an instructional video on a nearby tablet). To address this challenge, we develop step-wise recommendation models using a benchmarking dataset from public recipe data (with multiple recipe preparation steps) and employ human judges to provide recommendation labels for each step. We show how variations of recurrent neural networks (RNNs) trained and evaluated on that dataset can provide step-wise recommendations (e.g., for online resources, companion devices, actions, etc.) at every step of the task, using different contextual information for prediction purposes. Beyond task histories, which are well explored in recommendations research [2], using the information from future steps (what the user will do *next*, which is known when steps are enumerated) to provide recommendations for the current step is a novel aspect of this work.

The paper makes the following research contributions:

- Introduce *step-wise recommendation* as a new challenge in helping people tackle complex tasks, in our case focused on multi-device and multi-modal interactions in recipe preparation.
- Develop machine learned models for step-wise recommendation that tackle different prediction tasks and use different contextual information (i.e., past, present, and/or future steps).
- Show that our recurrent neural network (RNN) model performs strongly at a range of prediction tasks/contexts and significantly outperforms a linear baseline (logistic regression) according to paired sample t-test, ($p < 0.05$).
- Present implications of these findings for the design of task support in digital assistants, recommender systems, and intelligent systems in general.

The remainder of the paper is structured as follows: Section 2 outlines related work in areas such as recommendation, task assistance, and multi-device experiences. Section 3 describes the MDX setup used to frame our recommendation problem. Section 4 defines step-wise recommendation in this setting and in general. The results are presented in Section 5, and are discussed along with their implications in Section 6. We conclude in Section 7 with a summary of our findings and pointers to future work.

2 BACKGROUND

Research in several areas is relevant to the work described in this paper, including contextual recommendation systems, task assistance (in general and in the cooking domain), and utilizing multiple devices to support task completion.

2.1 Recommendation

There has been considerable research on recommendation systems, based on collaborative filtering and consideration of user profiles [23, 39, 48]. While the specific details vary from work to work, context can broadly be defined as the set of characteristics and features that are considered for making recommendations. User histories collected over time can yield useful information about interests and preferences [15, 59]. Item-to-item recommendations have been proposed as a way to address the absence of user profiles or histories [55]. The use of contextual signals, such as temporal sequencing [34] and physical location [47], to dynamically adapt suggestions has also been explored in depth. Perhaps most relevant to our work is research on *session-based* recommendations [56], where the local task context is used to generate suggestions relevant to the current point in time. As with our study, RNNs have been successfully applied for session-based recommendations [25, 62]. However, in prior work, the session data typically only comprise time-bounded sequences of previous user actions (e.g., clicks, views) not the content of the task steps with their associated semantics nor the content of future steps, as we utilize in this study. The "personalized next-step recommendation framework" [46] is another example of research on contextual and adaptive recommendation, where the goal is to predict the next page that users are likely to spend significant time on. Step-wise recommendation for multi-step tasks, which we introduce in this paper, was not explored in that research.

2.2 General Task Assistance

Task completion is an important issue and many best practices and solutions have been proposed [4, 14]. Enabling intelligent systems to directly offer task assistance requires a solid understanding of user goals and intentions [27, 61] and their current task state [27, 32]. Dialog management tools (e.g., RavenClaw [8]) have been used to build task-support-oriented dialog systems that offer guided task completion [3, 9]. Conversational agents can help individuals complete complex activities such as open-ended data science tasks [20] or form filling on the web [5, 28], and coordinate task completion between individuals [42, 64]. There has been additional research on creating multi-modal tools with more capability in deducing user status than conventional conversational assistants [32, 53]. More broadly, intelligent tutoring systems can follow students' reasoning and provide feedback on errors at every step in the learning process [40]. Step-by-step tutorials can be automatically derived from user demonstrations [13] or from community-generated videos [66]. Chang et al. [12] explored the use of voice interactions with video tutorials to help people navigate those videos as they attempted complex tasks. None of the aforementioned research in the dialog community or tutoring systems literature addresses the need for step-wise recommendation for complex sequential task support, especially across multiple devices.

2.3 Multi-device Task Support

The use of multiple digital devices to support people's activities has long been of interest to researchers [68]. The majority of United States residents own multiple digital devices. This fact makes multi-device task support (or MDXs, introduced earlier), spanning multiple devices simultaneously, viable for many individuals. The immediate advantage is the ability to leverage the unique strengths in each device to help people complete tasks [18, 58]. Devices can be used sequentially or in parallel, with the latter being more common in current practice [29]. The allocation of aspects of the complex tasks to devices can be based on many requirements, including form factors, functionalities, and convenience [16, 29, 54].

Returning to the cooking domain, as part of our work on MDXs, we presented a guided task completion system that used the capabilities of different devices *simultaneously* to drive task completion [69]. MDXs differ from cross-device experiences (CDXs) discussed in the literature on interaction design, human factors, and pervasive and ubiquitous computing [18, 58]. CDXs mainly focus on scenarios such as commanding (remote control), casting (displaying content from one device on another device), and task continuation (pausing and resuming tasks over time). Devices in CDXs are often used sequentially based on their suitability and availability, but in MDX scenarios, devices are used simultaneously, complementing one another's capabilities and addressing shortfalls in what individual devices can do alone. For instance, although a tablet/laptop device might have speaker and microphone, they might not be able to sufficiently support the needs that a far-field speaker can address, especially in a more noisy environment and/or when the user is far from the microphone (as might be the case in a cooking scenario, for example). A significant advantage of MDXs is that people can get immediate support by enabling users to pull together devices that they already own [69].

2.4 Task Assistance for Recipe Preparation

Recipe preparation is a good example of a common scenario in daily life in which people benefit from having an assistant which can guide them through the multiple steps of the recipe. Presence of multiple smart devices in kitchen spaces also makes it an excellent example of how MDX task support can be provided to users. Smart speakers are often placed in the kitchens of homes [44]. Digital assistants are often used for aspects of cooking such as setting timers or managing related processes [21]. In their current form, these assistants help users follow the step-by-step instructions that comprise recipes [51, 60]. Early discourse models in the recipe domain (e.g., [22]) have extended into coaching scenarios, where the user is guided during the preparation of a recipe using spoken dialog [38, 45]. Beyond direct spoken dialog commands and questions, Vtyurina and Fourney [65] demonstrated the importance of implicit verbal cues in the design of guided task experiences (e.g., "yup" or "alright" as an implicit request for the next step) in the cooking domain. Intelligence can also extend beyond task guidance, e.g., mining instructions from recipe pages which lack semantic markup to broaden the reach of task support [30] or identifying recipe refinements from user comments in online recipes [19].

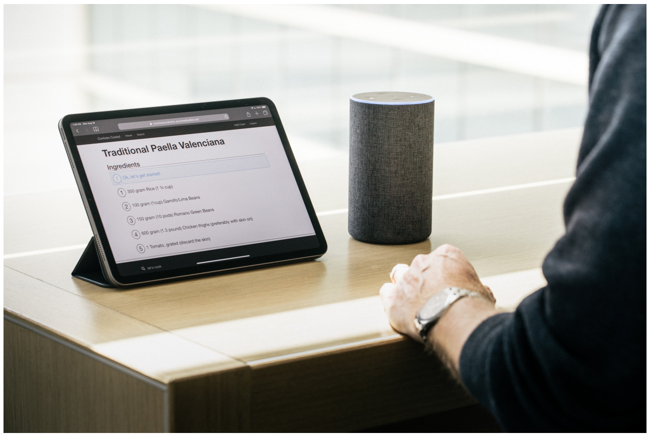


Figure 2: Multi-device support with the AskChef application built on our MDX framework, showing a smart speaker and a tablet device being used simultaneously.

2.5 Contributions Over Previous Work

Our research makes several contributions over prior work. First, although the use of context and user history to generate recommendations has been studied extensively, focusing on complex tasks and using user history *and* future steps to help with the current task step has not been as well studied. Session-based recommendations have used prior events rather than semantically-rich previous steps in the task. Second, research on task assistance typically focuses on the task holistically rather than helping users in a step-wise manner, which enables the provision of targeted recommendations at the right time. Third, task assistance in the cooking domain focuses on guiding users through the steps of a task rather than providing additional support (digital resources, devices, etc.) to augment the task completion experience. Finally, prior work on multi-device support designs end-to-end experiences using multiple devices, which are known to the system in advance, rather than recommending devices that could best help the user with their current step.

Before describing our step-wise recommendation methods, we first briefly describe the *AskChef* cooking application, in which we ground our research on step-wise recommendation, and the MDX framework on which that application was developed.

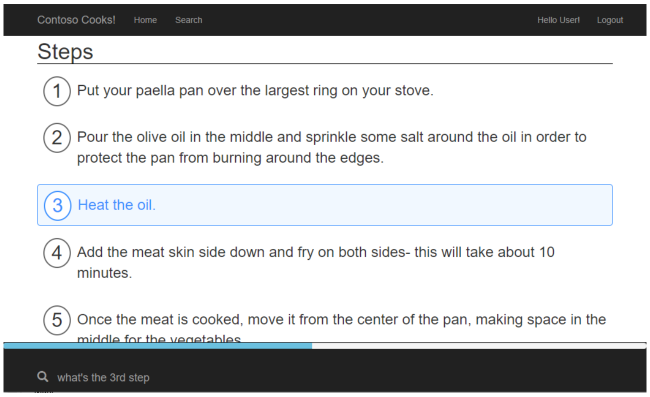
3 ASKCHEF

We envisage deploying step-wise recommendations in a guided task completion scenario similar to *AskChef* [43, 69]. AskChef is a cooking application built on top of a generic framework we have developed to enable developers to create their own MDXs.

The MDX framework consists of three layers: (1) system, (2) intelligence, and (3) user experience (UX). The *System* layer provides the necessary infrastructure to manage authentication, state management (both dialog context and navigational context), parsing of web page content (in [43] this focused on web pages that use schema.org semantic markup), and synchronization across devices via an event hub. The *Intelligence* layer provides the models necessary for language understanding (recognizing the intent of, and responding to, user utterances), document understanding, question understanding, and answer generation. Domain models apply to the specific domain supported in the experience (cooking in the case of AskChef). Horizontal models span domains and offer support for more general activities such as navigation (scrolling, pagination, etc.). Finally, the *UX* layer provides the visual presentation logic for visualizing the current state. The visual UX layer can be applied on top of third-party web content (e.g., via a web browser extension) or integrated into first-party experiences directly. In the case of cooking, the UX layer presents the recipe, highlights the current step, and provides a visual readout of spoken responses.

The MDX framework allows the deployment and synchronization of multiple devices for task assistance. For example, in the case of the AskChef application, a smart speaker, such as a Amazon Echo or Google Home, is paired with a screen-capable device such as PC, tablet, or smartphone to add a visual interface to the experience; see Figure 2. This enables users to quickly and easily recognize where they are within a given task. To support such a system, a broker component sits between the web page content and backend services (i.e., smart speaker skill, intent understanding, question-answering models, etc.). The broker exposes an application programming interface (API) which provides synchronization of actions between the voice and visual interfaces, as well as enabling visual cues to the user (e.g., step-by-step highlighting) indicating where they are within a task. This can be especially useful given the well-known limitations of short-term memory [41] and associated difficulties in keeping track of task state when engaged in a complex task.

As mentioned earlier, Figure 1 shows an example of typical cooking setup in AskChef, where a participant in a user study [43] receives multi-device support via a smart speaker and laptop. In the implementation of AskChef used by Nouri et al. [43], a web browser plugin was employed to unobtrusively overlay task guidance on existing recipe web sites. Figure 3 shows an alternative implementation (a closeup of the experience shown in the screen in Figure 2) whereby the guided task support from AskChef (e.g., highlight step and/or answers questions) is offered via a dedicated user experience, separate from the underlying first-party website (Contoso Cooks), with larger font (visible from a distance) and clearer information about the current step and task progress (through an on-screen progress bar near the bottom of Figure 3).



Steps

1 Put your paella pan over the largest ring on your stove.

2 Pour the olive oil in the middle and sprinkle some salt around the oil in order to protect the pan from burning around the edges.

3 Heat the oil.

4 Add the meat skin side down and fry on both sides- this will take about 10 minutes.

5 Once the meat is cooked, move it from the center of the pan, making space in the middle for the vegetables

what's the 3rd step

Figure 3: Screenshot of the AskChef application visual interface for first-party content (in this case an internal cooking website created for demonstration purposes). In this example, the system has highlighted the current step (Step 3) in response to a user voice request ("what's the 3rd step?").

Interactions with AskChef take two forms: (1) *Voice commands* used for navigation within a task (e.g., asking "what is the next step?") and open-ended question-and-answer exchanges specific to the task, and (2) *Touch commands*, either via mouse or finger(s) used for navigation within a specific task. AskChef updates the task state or responds with an appropriate answer when users pose questions regarding the task. The cooking task is considered completed once users reach the last step in the recipe. It is during such step-by-step progress through a complex task that an intelligent system could offer step-wise recommendations to users that consider the current step in the task and prior/future steps.

4 STEP-WISE RECOMMENDATION

Step-wise recommendation is a challenging activity that aims to align content of a complex task (recipe information, instructions, and support in the context of the AskChef application introduced in the previous section) with world knowledge related to the task (task progression status and multi-modal contextual features sensed from the environment) and user state. Our goal is to provide suitable recommendations to users at different steps throughout the complex task during the use of a multi-device DA. Information regarding the task (e.g., recipe steps) is already known to the DA, however, consideration of the set up and the dynamics of the interaction between the user and the system, real-world contextual updates add a challenging level of complexity to what the recommendation prediction model should do. To facilitate our research on step-wise recommendation, we created a novel and semantically rich benchmarking dataset that is used to train and test our prediction models and could be used for a number of purposes beyond step-wise recommendation. Sections 6 and 7 mention examples of further applications that are suitable for this benchmark dataset.

4.1 Data

Data collection for creation of our benchmark dataset for step-wise recommendation was performed in two steps which are described in the following subsections.

4.1.1 Collection of the Recipe Dataset. Given our focus on the cooking domain, recipes are the main source of data for our recommendation models. These recipes are extracted from websites which contain content under Creative Commons attribution license. In total, we collected 16,659 recipes from Foodista.com. These recipes contain the following information: recipe name and identification number, recipe yield (number of people that the recipe serves), ingredients, and instructions. We used a subset of recipes from this source to create tasks on a crowdsourcing platform to obtain recommendation information and annotations from human judges. Recipes were selected based on the quality of the recipe content (subjectively assessed by the authors), language (narrowing it down to English language recipes) and ensuring that each had 5-7 steps (this was a reasonable balance between being suitable for human labeling in a short timeframe, while still having sufficient in-task data for recommendation purposes).

4.1.2 Collection of the Recommendation Labels on Crowdsourcing Platform. We conducted our data collection for our recommendation benchmark dataset on a crowdsourcing platform developed within our organization which uses external judges from vendors such as clickworker.com. This platform connects human intelligence tasks (HITs) with a large population of crowdworkers in many geographic locales. It allows specification including participation according to the country of residence and native language (US and English in our case), and for limiting the maximum number of tasks done by a single worker. Crowdworkers can only work on the HITs if they meet the requirements and read the HIT guidelines.

The HIT instructions informed the crowdworkers that the goal was to make a food recipe step by step, while interacting with smart devices (e.g., smart speaker plus a device with a display, as in Figure 1). The crowdworkers were informed that the devices provide recommendations and feedback for preparation of the recipe. A total of 421 HITs were published resulting in annotation of 91 randomly-selected recipes with on average 5.31 steps per recipe.

Each recipe was shown in its entirety with information on the title of the recipe and its ingredients. The HIT was designed to dynamically enumerate and populate a table of questions for every step in the recipe. Figure 4 (overleaf) has for an example of one of the HITs used in the study. A set of questions are interactively shown for every step of the instructions of the recipe and one set of questions is asked about the entire recipe at the end of the HIT.

The specific information captured about each step was:

1. **Show suggestion:** A binary label (yes/no) indicating whether the judge recommends that a suggestion be displayed.
2. **Suggestion format:** The desired format of the suggestion (e.g., video, text).
3. **Action to take:** The desired action to take upon determining the suggestion format (e.g., show image. show video)
4. **Device to use:** The desired delivery mode of the suggestion (e.g., tablet, speaker).

Following the data collection, all data collected from crowdworkers were reviewed by one of the authors (EN) and invalid or low-quality entries were removed. This manual quality check resulted in rejection of 13.3% of the collected data. Table 1 shows the distributions of the discrete labels in the dataset. Some steps may receive multiple labels for the format, action, and device judgments.

Crowdsourcing was selected as the acquisition strategy for the recommendation labels. This allowed us to obtain a reasonable number of labels quickly and at low cost. We assumed the crowdworkers would be highly-familiar with the cooking scenario, allowing them to provide high-quality assessments on resources and devices that could be helpful for completing the cooking task. This is certainly reasonable as a starting point for training our recommendation models. In practice, we would seek to further bootstrap the dataset with data collected when the system is deployed in production.

Table 1: Discrete label distributions across all steps in the recipes provided to crowdworkers.

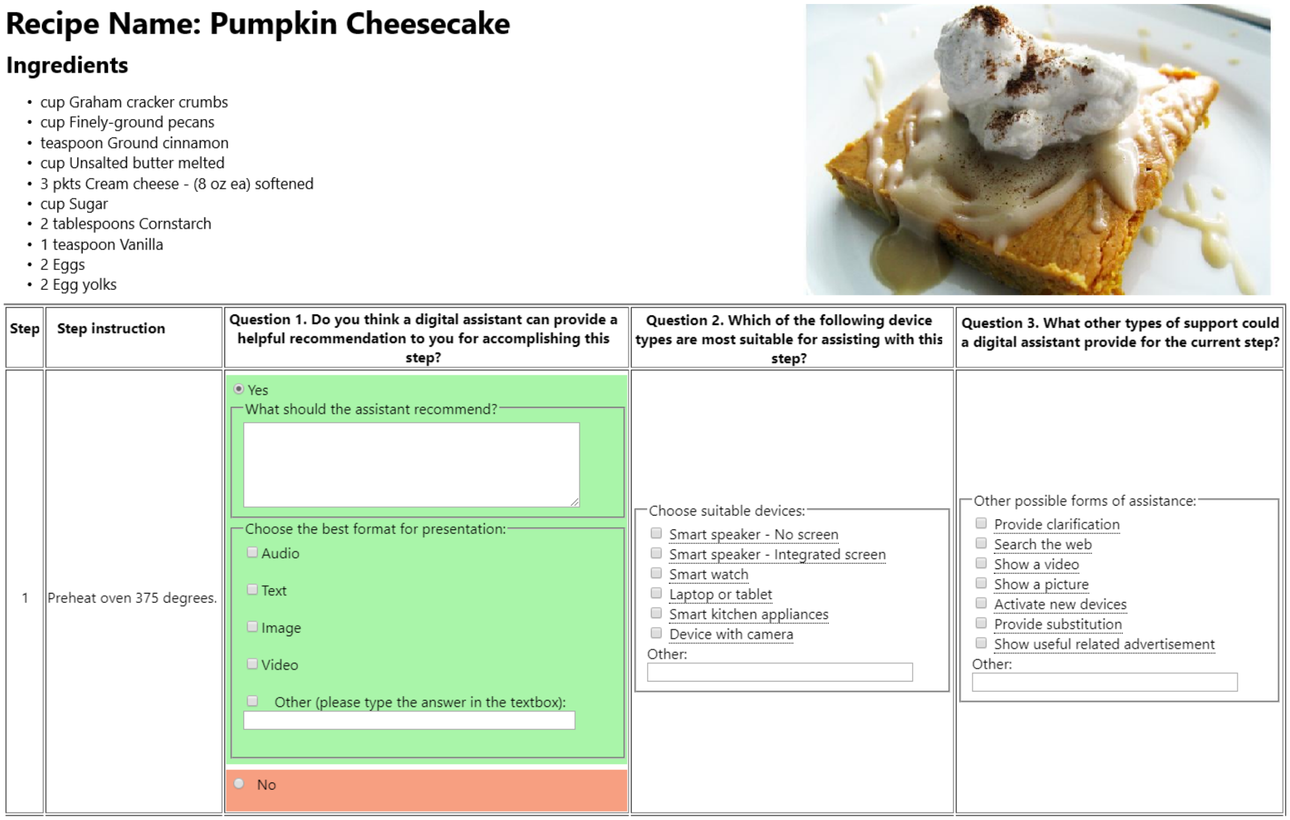
Label	Step Count	% of Label
Show suggestion:		
Yes	1860	17.6%
No	8711	82.4%
Suggestion format:		
Video	925	33.6%
Text	671	24.4%
Audio	600	21.8%
Image	554	20.1%
Action to take:		
Image	3250	26.3%
Video	3193	25.8%
Activate	1460	11.8%
Clarify	1349	10.9%
Substitution	1219	9.9%
Search	1217	9.8%
Show advertisement	668	5.4%
Device to use:		
Tablet	4038	31.5%
Wearable	2274	17.7%
Speaker with screen	2083	16.2%
Speaker without screen	1995	15.6%
Smart appliance	1652	12.9%
Device with camera	782	6.1%

4.2 Step-wise Recommendation Models

Our step-wise recommendation models consist of prediction models in form of binary or multi-label classification, depending on the prediction task. Our recommendation system makes four decisions at every step of the task in order to provide the appropriate information to the user. The following sections explain the details of what decisions are made and defines the prediction tasks for step-wise recommendation.

4.2.1 Step-wise Decision on Recommendation. This is the primary decision in our recommendation pipeline and involves making the decision about whether or not a recommendation should be presented to the user. The ideal system can distinguish between points during a task that a recommendation should be provided versus when only conversation regarding the task instructions should be handled. This is a critical decision for recommender systems in general since it has been shown that undesired recommendation can cause unnecessary distraction [24, 52].

In our research, this decision is cast as a binary classification problem, determining whether or not the user at the current step of the task would benefit from additional information. For example, in Figure 3, steps 1 and 3 are straightforward and might not require recommendations, whereas steps 2 and 4 are more complex and might benefit from additional content (e.g., an instructional video).



Recipe Name: Pumpkin Cheesecake

Ingredients

- cup Graham cracker crumbs
- cup Finely-ground pecans
- teaspoon Ground cinnamon
- cup Unsalted butter melted
- 3 pkts Cream cheese - (8 oz ea) softened
- cup Sugar
- 2 tablespoons Cornstarch
- 1 teaspoon Vanilla
- 2 Eggs
- 2 Egg yolks

Step	Step instruction	Question 1. Do you think a digital assistant can provide a helpful recommendation to you for accomplishing this step?	Question 2. Which of the following device types are most suitable for assisting with this step?	Question 3. What other types of support could a digital assistant provide for the current step?
1	Preheat oven 375 degrees.	● Yes What should the assistant recommend? Choose the best format for presentation: □ Audio □ Text □ Image □ Video □ Other (please type the answer in the textbox): ○ No	Choose suitable devices: □ Smart speaker - No screen □ Smart speaker - Integrated screen □ Smart watch □ Laptop or tablet □ Smart kitchen appliances □ Device with camera Other:	Other possible forms of assistance: □ Provide clarification □ Search the web □ Show a video □ Show a picture □ Activate new devices □ Provide substitution □ Show useful related advertisement Other:

Figure 4: A screenshot showing an example of the HIT developed to collect judgments on the steps in the recipe.

4.2.2 Step-wise Decision on Recommended Format. If a suggestion is deemed appropriate for the current step in the task, a subsequent decision would be made regarding the format of the recommendation that is to be provided. In the HIT, the following choices are shown after crowdworkers select "yes" for whether a recommendation is appropriate: "Audio," "Text," "Image," and "Video." Crowdworkers can also enter any other format. This decision is cast as a multi-class prediction problem.

4.2.3 Step-wise Decision on Recommended Device. In the context of multi-device usage (as with AskChef), multiple devices with varying capabilities are synchronized together to assist the user. Once the system has made the determination as to whether or not to provide a recommendation and selects the most appropriate format and action for the type of recommendation content, the system needs to determine which available device is most suitable for presenting this information to the user. The choices are "Smart speaker - No screen," "Smart speaker - Integrated screen," "Smart watch," "Laptop or tablet," "Smart kitchen appliances," and "Device with camera." Crowdworkers can also enter any other device. As with recommended format and recommended action, this decision is cast as a multi-class prediction problem.

4.2.4 Step-wise Decision on Recommended Action. When the intelligent system wants to provide a recommendation, once the system has made the determination as to whether or not to provide a recommendation and selects the most appropriate format for the type of the recommendation content, the system needs to determine which action would be most suitable for presenting this information to the user. The choices for possible types of assistance are "Provide clarification," "Search the web," "Show a video," "Show a picture," "Activate new devices," "Provide substitution," "Show useful related advertisement," and crowdworkers can also enter any other action. This decision is also a multi-class prediction problem.

4.3 Recommendation Models

To address the prediction tasks introduced in previous section, we used several common machine learning algorithms to perform the recommendation modeling. We report the results from baselines, linear models, and neural models for all of our prediction tasks.

To run the prediction models, we split the dataset into train, validation, and test sets with the ratio of 0.7, 0.1, and 0.2 respectively. We use classification accuracy as the main metric for evaluating the accuracy of our prediction models [24], which is simply the percentage of predictions made by the classifier that are correct.

4.3.1 Baseline Models. Two baselines: (1) a *random* classifier that randomly assigns class labels to instances, and (2) a *marginal* classifier that always assigned the dominant class labels to instances.

4.3.2 Logistic Regression Model. We use logistic regression (LR) a more sophisticated baseline. Logistic regression is an interpretable model that has successfully been used for text classification. This model is compact and cost-efficient for training. We use word level unigrams combined with term frequency - inverse document frequency (TF-IDF) to represent text features.

4.3.3 Neural Models. We use neural models based on Recurrent Neural Networks [57] for both single-class and multi-class short text classification. RNNs' sequential architecture allows it to exhibit temporal behavior and capture sequential data and therefore it has become a natural approach when dealing with textual data [33]. A long-short-term-memory LSTM [26] network is used for our RNN network, with 4 bidirectional layers. A single fully connected layer is employed on top of bi-directional LSTM. We experimented with using the following two conditions: by using the last hidden state and using the average of all hidden states as the input to the fully-connected classification layer. A trainable word embedding vector is employed as input layer under two conditions: with both random initialization and pre-trained model of GloVe embeddings [49]. The goal is to evaluate the advantage of using pre-trained models over random initialization. Grid-search was performed for choosing the dimension of the Glove word embedding and the results reported are for dimension size 300. We performed five-fold cross validation for assessment of our models. The results of our experiments are reported in detail in the next section.

5 EXPERIMENTS AND RESULTS

In this section, we describe the experimental methodology and present the results of our experiments. For each of the prediction tasks that we have already defined in Section 4.2, we study how the various recommendation models (random and marginal baselines, logistic regression, and neural network) perform under three distinct conditions:

(1) **Recommendation in the moment**: Consideration of the information from the current step of the task (i.e., the step that the user is currently executing).
(2) **Recommendation based on current step + past history**: Consideration of current step plus additional information from the previous steps in the task.
(3) **Recommendation based on current step + future planning**: Consideration of current step plus additional information from the future steps in the task.

The reason that we are studying these three conditions is the assumption that contextual information (including future steps, often unavailable for many recommendation scenarios, but available in this setting) is a useful basis on which to generate recommendations tailored to the current situation. This is the major distinction in how our recommendation model operates as opposed to conventional models, which are commonly cast as a problem of ranking among available alternatives. In this section, we review the findings from our experiments in applying the prediction models.

5.1 Decision on Recommendation

This decision is cast as a binary classification problem, determining whether or not recommendation is needed at each step. Table 2 shows the accuracy of our models for deciding whether or not the assistant should provide a recommendation to the user. The accuracy metrics are computed across all steps in the test set (20% of the labeled dataset). Based on the results of our experiments, we can see that prediction accuracy for the logistic regression model shows improvement in performance once **history** and **future** information (see Section 5 for a definition) are considered by the prediction model, increasing from 55.1% to 77.6% and 77.3% respectively, with considering the history achieving the best performance among all. In the case of the neural network models, consideration of the future steps results in statistically significant improvements (according to paired sample t-test, $p < 0.05$) in model accuracy for both versions of the RNN model based only on the current step, increasing from 87.5% to 93.0%, and RNN combined with GloVe increasing from 88.7% to 93.4%.

5.2 Decision on Recommended Format

As described in Section 4.2.2, if the decision to provide a recommendation is affirmative, a subsequent decision would be made regarding its format (video, image, etc.). The results for this task are reported in Table 3. Based on the results of our experiments, prediction accuracy for the logistic regression model shows improvement in performance once **history** and **future** information are considered by the prediction model, increasing from 73.4% to 81.3% and 81.8% respectively, with the consideration of history information achieving the highest performance among all. In the case of the neural network models, considering the future steps results in statistically significant improvements in model accuracy for both versions of the RNN model is achieved over the LR model and the baselines (according to paired sample t-test, $p < 0.05$), increasing from 91.9% to 95.1%. RNN combined with GloVe increased model accuracy from 94.4% to 95.7%. The addition of semantic representations such as GloVe appears to benefit the RNN models considerably. It may be that semantics are closely connected to the best format to represent (and hence, perhaps, support) the current task step.

5.3 Decision on Recommended Device

In the context multi-device usage, multiple devices with varying capabilities are synchronized together to assist the user. After the system determines to provide a recommendation and selects the most appropriate format and action, it also needs to determine which available device would be most suitable for presenting this information to the user. The results of our prediction model for this decision are shown in Table 4. Based on the results of our experiments, prediction accuracy for the logistic regression model shows an improvement in performance once **history** and **future** information are considered, increasing from 73.8% to 83.4% and 83.1% respectively. For the neural network models, considering the future steps leads to significant improvements in performance for both versions of the RNN model (according to paired sample t-test, $p < 0.05$), increasing from 84.8% to 86.3%. For RNN combined with GloVe, model accuracy increases from 82.8% to 83.3%. As with the decision on recommended format, semantics may be highly connected to the device type.

5.4 Decision on Recommended Action

Once the system has made the determination to make a recommendation it chooses the most appropriate form of action to suggest to the user. The results for this decision are shown in Table 5.

Table 2: Accuracy of models for decided whether a recommendation is needed. Models which significantly outperform both baselines (according to paired sample t-test, $p < 0.05$) are denoted with (*). Best performing models for each condition are shown in bold.

Condition	Only current step	Previous steps + current step	Current step + future steps
Random	50.00	50.00	50.00
Marginal	82.40	82.40	82.40
Logistic Regression	55.10	77.60	77.30
RNN	87.50 *	84.37 *	92.96 *
RNN+GloVe 300	***88.67*** *	***89.45*** *	***93.36*** *

Table 3: Accuracy of models for deciding recommended format. Models which significantly outperform both baselines (according to paired sample t-test, $p < 0.05$) are denoted with (*). Best performing models for each condition are shown in bold.

Condition	Only current step	Previous steps + current step	Current step + future steps
Random	25.00	25.00	25.00
Marginal	33.64	33.64	33.64
Logistic Regression	73.37 *	81.29 *	81.77 *
RNN	91.90 *	***93.30*** *	95.10 *
RNN+GloVe 300	***94.40*** *	89.20 *	***95.70*** *

Table 4: Accuracy of models for deciding recommended devices. Models which significantly outperform both baselines (according to paired sample t-test, $p < 0.05$) are denoted with (*). Best performing models for each condition are shown in bold.

Condition	Only current step	Previous steps + current step	Current step + future steps
Random	16.67	16.67	16.67
Marginal	31.49	31.49	31.49
Logistic Regression	73.80 *	83.43 *	83.12 *
RNN	***84.80*** *	***89.70*** *	***86.30*** *
RNN+GloVe 300	82.80 *	89.30 *	83.30 *

Table 5: Accuracy of models for deciding recommended actions. Models which significantly outperform both baselines (according to paired sample t-test, $p < 0.05$) are denoted with (*). Best performing models for each condition are shown in bold.

Condition	Only current step	Previous steps + current step	Current step + future steps
Random	14.28	14.28	14.28
Marginal	26.30	26.30	26.30
Logistic Regression	79.36 *	85.93 *	84.97 *
RNN	87.60 *	89.00 *	***92.40*** *
RNN+GloVe 300	***88.20*** *	***91.10*** *	92.00 *

The experimental results show that prediction accuracy for the logistic regression model improves once **history** and **future** information are considered, increasing from an accuracy of 79.4% to 85.9% and 85.0% respectively. In the case of the neural network models, considering the future steps results in statistically significant improvements (according to paired sample t-test, $p < 0.05$) in performance for both versions of the RNN model based on the current step, increasing from 87.6% to 92.4%. RNN combined with GloVe increases model accuracy from 88.2% to 92.0%. The RNN model results in highest performance for all three conditions.

6 DISCUSSION AND IMPLICATIONS

This study has introduced step-wise recommendation and shown that it is feasible to generate reasonable recommendations for the current step in a complex task, using data from that step plus other information in the task (past or future). Our study shows that RNN-based approaches perform well at the task and that these models can be enhanced through the addition of semantic information via word embeddings. We also show that by adding the context (additional steps from the same task) the accuracy of the predictions improves. Future steps help for some prediction tasks (whether a recommendation is required and format) but not for others (actions and devices). One explanation for this is that actions and devices are encoded in preceding steps of the task and that future steps denote a change in direction, where new actions or devices are required, that can in fact lessen prediction performance.

The findings are promising for the development of task support to help surface the additional information to people at the right time. This relates to research on just-in-time information access [10]. Although the study focused on the cooking domain, there are many other application domains containing complex tasks, including home and auto repair, furniture assembly, and calendar management. The study was performed in the context of multiple device usage, which unlocks a range of new interactive experiences to help users complete complex tasks. As the results of our study show, the methods could still be applied in single-device settings where there are no companion devices, e.g., for determining the best format for the current step or useful actions.

There are a few limitations that we should acknowledge. One limitation is the size of the dataset used in the study. More work is needed to collect a larger dataset and understand the effects of data volume on the accuracy of the predictions. Another is the nature of the data used, which is limited to the text of the steps in the task (from which semantic representations [namely word embeddings and pretrained models] can be derived). An ideal recommendation model should also consider data from user interactions and other related sources of information that integrate the world knowledge related to the task. Currently, our main source of data for training the recommendation models is the information from recipes extracted from online sources. This is primarily because of our limitations in collecting a scalable amount of data of users using our AskChef application to cook actual recipes. One solution that we are investigating is the possibility of collecting usage data on a crowdsourcing platform by imposing the requirement that crowdworkers have at least a screen and speaker and perhaps even perform physical tasks such as recipe preparation. Another solution is alignment of other sources for world knowledge (e.g., Wikipedia pages and recipe cooking video content on the web) which is related to our task with our recipe dataset. A final limitation is a lack of access to the context of use, e.g., which devices are available to the user at the time of the task, which is an important factor to consider during recommendation generation.

Since this is the first study of step-wise recommendation for complex task support, there are many avenues for future work. This includes considering the interactions and dependencies between different steps in the task rather (e.g., co-references, etc.) rather than only considering them only as blocks of text. In this study, we considered each of the prediction tasks independently. An interesting angle for future work is to leverage research on multi-target learning [7] and multi-task learning [11], to allow interactions between different prediction outcomes to be considered (e.g., predicting the format/action/device is not required if no recommendation is needed). Finally, it is important to explore the development of generative models (e.g., [36, 63]) to create content for the recommendation [6].

7 CONCLUSIONS AND FUTURE WORK

This paper has introduced step-wise recommendation as a way to support users engaged in complex tasks. We focused on the cooking domain and a multi-device setting, given the complexity of the tasks, the availability of data, and the richness of the recommendation possibilities. We built deep learning models for a range of prediction tasks: whether a recommendation is needed, what format the suggestions should take, what actions should be taken, and which device should be used. Our deep learning models outperformed our baselines and a logistic regression model, and the results showed that the addition of semantic information about the preceding and succeeding steps yielded further gains in prediction accuracy. As part of the future work we will investigate the performance of the transformer based text encoder models such as BERT [17], RoBERTa [35] and XLNet [70] for recommendation modeling. In addition, future work is needed to explore alternative application scenarios and domains. There are also a range of important follow-ups in considering collecting additional data and building more sophisticated recommendation algorithms that account for the context of use (e.g., available devices that the user has permission to use) and more fully model the relationships between the current step and the other steps in the complex task.

REFERENCES

[1] 2017. Alexa, Say What?! Voice-Enabled Speaker Usage to Grow Nearly 130% This Year. https://www.emarketer.com/Article/Alexa-Say-What-Voice-Enabled-Speaker-Usage-Grow-Nearly-130-This-Year/1015812

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender Systems Handbook*. 217–253.

[3] G Aist, J Dowding, BA Hockey, M Rayner, J Hieronymus, D Bohus, B Boven, N Blaylock, E Campana, S Early, et al. 2003. Talking through procedures: An intelligent Space Station procedure assistant. In *EACL*. 285–295.

[4] David Allen. 2003. *Getting Things Done. The Art of Stress-Free Productivity.* Penguin.

[5] James Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom. 2007. Plow: A collaborative task learning agent. In *AAAI*. 1514–1519.

[6] Homanga Bharadhwaj, Homin Park, and Brian Y Lim. 2018. RecGAN: recurrent generative adversarial networks for recommendation systems. In *RecSys*. 372–376.

[7] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. 1998. Top-down induction of clustering trees. In *ICML*. 55–63.

[8] Dan Bohus and Alexander I Rudnicky. 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *EUROSPEECH*.

[9] Dan Bohus and Alexander I Rudnicky. 2005. LARRI: A language-based maintenance and repair assistant. In *Spoken Multimodal Human-computer Dialogue in Mobile Environments*. 203–218.

[10] Jay Budzik and Kristian J Hammond. 2000. User interactions with everyday applications as context for just-in-time information access. In *IUI*. 44–51.

[11] Rich Caruana. 1997. Multitask learning. *Machine Learning* 28, 1 (1997), 41–75.

[12] Minsuk Chang, Anh Truong, Oliver Wang, Maneesh Agrawala, and Juho Kim. 2019. How to design voice based navigation for how-to videos. In *SIGCHI*. 1–11.

[13] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: automatic generation of step-by-step mixed media tutorials. In *UIST*. 93–102.

[14] Stephen R. Covey. 2004. *The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change* (15 ed.). Free Press, New York.

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

[16] David Dearman and Jeffery S Pierce. 2008. It's on my other computer!: computing with multiple devices. In *SIGCHI*. 767–776.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[18] Tao Dong, Elizabeth F Churchill, and Jeffrey Nichols. 2016. Understanding the challenges of designing and developing multi-device experiences. In *DIS*. 62–72.

[19] Gregory Druck and Bo Pang. 2012. Spice it up?: Mining refinements to online instructions from user generated content. In *ACL*. 545–553.

[20] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S Bernstein. 2018. Iris: A conversational agent for complex tasks. In *SIGCHI*. 473.

[21] David Graus, Paul N Bennett, Ryen W White, and Eric Horvitz. 2016. Analyzing and predicting task reminders. In *UMAP*. 7–15.

[22] Nancy Green and Jill Fain Lehman. 2002. An integrated discourse recipe-based model for task-oriented dialogue. *Discourse Processes* 33, 2 (2002), 133–158.

[23] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR*. 230–237.

[24] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53.

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

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

[27] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. 1998. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *UAI*. 256–265.

[28] Rogers Jeffrey Leo John, Navneet Potti, and Jignesh M Patel. 2017. Ava: From data to insights through conversations. In *CIDR*.

[29] Tero Jokela, Jarno Ojala, and Thomas Olsson. 2015. A diary study on combining multiple information devices in everyday activities and tasks. In *SIGCHI*. 3903–3912.

[30] Chloé Kiddon, Ganesa Thandavam Ponnuraj, Luke Zettlemoyer, and Yejin Choi. 2015. Mise en place: Unsupervised interpretation of instructional recipes. In *EMNLP*. 982–992.

[31] Xinyu Lei, Guan-Hua Tu, Alex X Liu, Chi-Yu Li, and Tian Xie. 2017. The insecurity of home digital voice assistants-amazon alexa as a case study. *arXiv preprint arXiv:1712.03327* (2017).

[32] Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-seng Chua. 2018. Knowledge-aware Multimodal Dialogue Systems. In *MM*. 801–809.

[33] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101* (2016).

[34] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *ICDM*. 1053–1058.

[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 preprint arXiv:1907.11692* (2019).

[36] Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Neural text generation: past, present and beyond. *arXiv preprint arXiv:1803.07133* (2018).

[37] Ewa Luger and Abigail Sellen. 2016. Like having a really bad PA: The gulf between user expectation and experience of conversational agents. In *SIGCHI*. 5286–5297.

[38] Filipe Martins, Joana Paulo Pardal, Luís Franqueira, Pedro Arez, and Nuno J Mamede. 2008. Starting to cook a tutoring dialogue system. In *IEEE Spoken Language Technology Workshop*. 145–148.

[39] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. 2002. Content-boosted collaborative filtering for improved recommendations. In *AAAI*. 187–192.

[40] Douglas C Merrill, Brian J Reiser, Michael Ranney, and J Gregory Trafton. 1992. Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences* 2, 3 (1992), 277–305.

[41] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 63, 2 (1956), 81.

[42] Karen Myers, Pauline Berry, Jim Blythe, Ken Conley, Melinda Gervasio, Deborah L McGuinness, David Morley, Avi Pfeffer, Martha Pollack, and Milind Tambe. 2007. An intelligent personal assistant for task and time management. *AI Magazine* 28, 2 (2007), 47–47.

[43] Elnaz Nouri, Adam Fourney, Robert Sim, and Ryen W White. 2019. Supporting complex tasks using multiple devices. In *WSDM 2019 Workshop on Task Intelligence*.

[44] Sheryl Ong and Aaron Suplizio. 2016. Unpacking the Breakout Success of the Amazon Echo. https://www.experian.com/innovation/thought-leadership/amazon-echo-consumer-survey.jsp

[45] Joana Paulo Pardal and Nuno J Mamede. 2011. Starting to cook a coaching dialogue system in the olympus framework. In *Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop*. 255–267.

[46] Zachary A Pardos, Steven Tang, Daniel Davis, and Christopher Vu Le. 2017. Enabling real-time adaptivity in MOOCs with a personalized next-step recommendation framework. In *Learning@Scale*. 23–32.

[47] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. 2007. Location-based recommendation system using bayesian user's preference model in mobile devices. In *UbiComp*. 1130–1139.

[48] Michael J Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13, 5-6 (1999), 393–408.

[49] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*. 1532–1543.

[50] Sarah Perez. 2018. 39 million Americans now own a smart speaker, report claims. https://techcrunch.com/2018/01/12/39-million-americans-now-own-a-smart-speaker-report-claims/

[51] Emma Persky. 2017. Now we're cooking - the Assistant on Google Home is your secret ingredient. https://www.blog.google/products/assistant/cooking-with-theassistant-google-home-your-secret-ingredient/.

[52] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender Systems Handbook*. Springer, 1–35.

[53] Amrita Saha, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *AAAI*.

[54] Stephanie Santosa and Daniel Wigdor. 2013. A field study of multi-device workflows in distributed workspaces. In *UbiComp*. 63–72.

[55] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms.. In *WWW*. 285–295.

[56] J Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *EC*. 158–166.

[57] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[58] Katarina Segerståhl. 2009. Crossmedia systems constructed around human activities: a field study and implications for design. In *INTERACT*. 354–367.

[59] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *SIGIR*. 909–912.

[60] AllRecipes Staff. 2016. Introducing a Cool New Way to Cook: Allrecipes on Amazon Alexa. http://dish.allrecipes.com/introducing-allrecipes-onamazon-alexa/.

[61] Ming Sun, Yun-Nung Chen, and Alexander I Rudnicky. 2016. An intelligent assistant for high-level task understanding. In *IUI*. 169–174.

[62] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Workshop on Deep Learning for Recommender Systems*. 17–22.

[63] Guy Tevet, Gavriel Habib, Vered Shwartz, and Jonathan Berant. 2018. Evaluating Text GANs as Language Models. *arXiv preprint arXiv:1810.12686* (2018).

[64] Carlos Toxtli, Andrés Monroy-Hernández, and Justin Cranshaw. 2018. Understanding chatbot-mediated task management. In *SIGCHI*. 1–6.

[65] Alexandra Vtyurina and Adam Fourney. 2018. Exploring the role of conversational cues in guided task support with virtual assistants. In *SIGCHI*. 1–7.

[66] Xu Wang, Benjamin Lafreniere, and Tovi Grossman. 2018. Leveraging community-generated videos and command logs to classify and recommend software workflows. In *SIGCHI*. 285.

[67] D. Watkins. 2016. Strategy analytics: Amazon, google to ship nearly 3 million digital voice assistant devices in 2017. https://www.strategyanalytics.com/strategyanalytics/news/strategy-analytics-press-releases/strategy-analytics-pressrelease/...

[68] Mark Weiser. 1991. The Computer for the 21 st Century. *Scientific American* 265, 3 (1991), 94–105.

[69] Ryen W. White, Adam Fourney, Allen Herring, Paul N. Bennett, Nirupama Chandrasekaran, Robert Sim, Elnaz Nouri, and Mark J. Encarnación. 2019. Multi-device digital assistance. *Commun. ACM* 62, 10 (2019), 28–31.

[70] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*. 5754–5764.

[71] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *CIKM*. 177–186.