

Situated and Interactive Multimodal Conversations

Seungwhan Moon*, Satwik Kottur*, Paul A. Crook†, Ankita De†, Shivani Poddar†
Theodore Levin, David Whitney, Daniel Difranco, Ahmad Beirami
Eunjoon Cho, Rajen Subba, Alborz Geramifard

Facebook
✉ simmc@fb.com

Abstract

Next generation virtual assistants are envisioned to handle multimodal inputs (*e.g.*, vision, memories of previous interactions, and the user’s utterances), and perform multimodal actions (*e.g.*, displaying a route while generating the system’s utterance). We introduce Situated Interactive MultiModal Conversations (SIMMC) as a new direction aimed at training agents that take multimodal actions grounded in a *co-evolving* multimodal input context in addition to the dialog history. We provide two SIMMC datasets totalling ~ 13 K human-human dialogs (~ 169 K utterances) collected using a multimodal Wizard-of-Oz (WoZ) setup, on two shopping domains: (a) furniture – grounded in a shared virtual environment; and (b) fashion – grounded in an evolving set of images. Datasets include multimodal context of the items appearing in each scene, and contextual NLU, NLG and coreference annotations using a novel and unified framework of SIMMC *conversational acts* for both user and assistant utterances.

Finally, we present several tasks within SIMMC as objective evaluation protocols, such as structural API prediction, response generation, and dialog state tracking. We benchmark a collection of existing models on these SIMMC tasks as strong baselines, and demonstrate rich multimodal conversational interactions. Our data, annotations, and models are publicly available.¹

1 Introduction

As virtual digital assistants become increasingly ubiquitous, they are expected to be embedded in the day-to-day life of users the same way a human assistant would. We thus envision that the next generation of virtual assistants will be able to process multimodal inputs and provide multimodal outputs beyond the traditional NLP stack. To this end, we present **Situated Interactive MultiModal Conversations (SIMMC)** tasks and datasets as a starting point in this new research direction. Specifically, SIMMC focuses on **task-oriented** dialogs that encompass a **situated multimodal context**, where situated implies that the user and assistant are continually co-observing the same context, and that context can be updated on each turn. We provide two new SIMMC datasets in the domain of interactive shopping, collected using the SIMMC Platform (Crook et al., 2019): (1) Furniture and (2) Fashion. Moreover, we provide fine-grained annotations to allow for both end-to-end and component-level modelling. The annotation includes natural language understanding (NLU), multimodal-coreference, multimodal state tracking, assistant actions, natural language generation (NLG), and item appearance logs.

Fig. 1 illustrates an exemplary dialog from our SIMMC-Furniture dataset, where a user is interacting with an assistant with the goal of browsing for furniture. In our setting, the assistant can update the co-observed environment to create a new multimodal context based on the preceding dialog with the user, *e.g.*, visually presenting recommended chairs in a virtual reality (VR) environment, or responding to the request “I like the brown one. *Show me the back of it.*” by executing the actions of *focusing on*, and *rotating* the indicated item. These actions change the shared multimodal context, which grounds the next

* Joint first authors. ✉ {shanemoon, skottur}@fb.com.

† Joint second authors. ✉ {pacrook, deankita, shivanip}@fb.com.

¹<https://github.com/facebookresearch/simmc>

This work is licensed under a Creative Commons Attribution 4.0 International License.

License details: <http://creativecommons.org/licenses/by/4.0/>

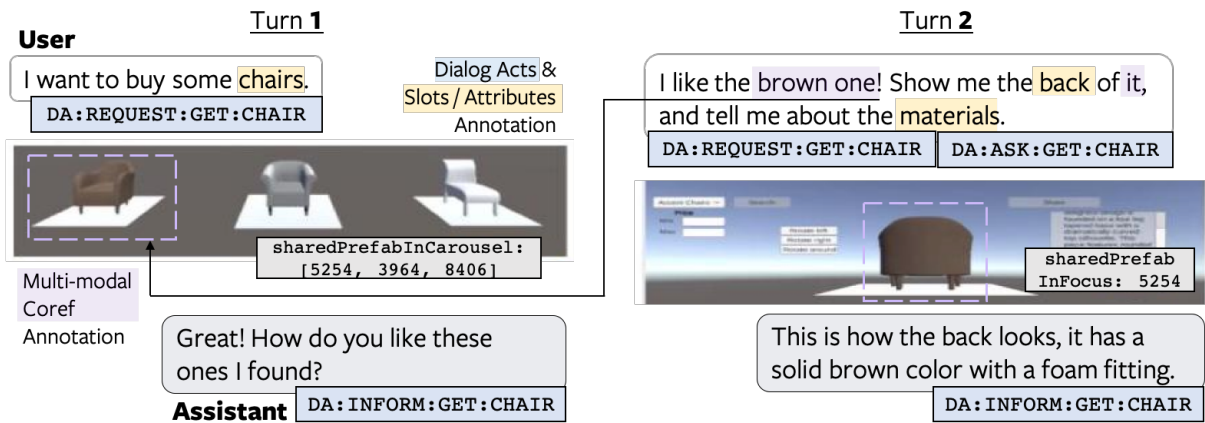


Figure 1: Illustration of a SIMMC dialog: a user and an assistant interact in a co-observed multimodal environment for a shopping scenario. The dialog is grounded in an *evolving* multimodal context. The ground-truth of which items (*e.g.*, prefabs) appear is known for each view.

part of the dialog. The example also highlights novel challenges such as multimodal action prediction (*italics* above) and multimodal coreference resolution (underlined elements above).

2 Novelty & Related Work

Tab. 1 presents main distinctions of SIMMC compared to the existing multimodal dialog datasets.

Multimodal datasets for co-observed, real-world assistant. With the ultimate goal of laying the foundations for the real-world assistant scenarios, we assume a co-observed multimodal context between a user and an assistant. This shifts the primary focus onto the core problem of grounding conversations in the co-observed multimodal context. In contrast, the existing literature (Das et al., 2017; Kottur et al., 2019; De Vries et al., 2017; de Vries et al., 2018), drawing motivation from the Visual Question Answering (Antol et al., 2015), posits the roles of a primary and secondary observer, *i.e.*, *questioner* and *answerer*, who do not co-observe the same multimodal context.

In addition, we study scenarios in which the situated multimodal context is dynamically updated, reflecting the agent actions. In our settings, agent actions can be enacted on both the object-level – changing the view of a specific object within a scene, and the scene-level – introducing a new scene or an image. While the dialog-based image retrieval tasks (Guo et al., 2018; Saha et al., 2018) and the visual navigation tasks (Thomason et al., 2019; de Vries et al., 2018) do comprise context updates, they are limited to the introduction of new visual scenes, *e.g.*, new images or locations.

Focus on task-oriented dialogs. We frame the problem as a *task-oriented*, multimodal dialog system, with the aim of extending the capabilities of digital assistants to real-world multimodal settings. While one focus area of the dialog community is on *task-oriented* dialog, which has practical applicability consumer-facing virtual assistants (Henderson et al., 2014; Budzianowski et al., 2018; Eric et al., 2019; Rastogi et al., 2019; Chen et al., 2020), this form of dialog is often neglected in many existing multimodal ‘dialog’ datasets (both in terms of the task design and the annotations), where the primary focus lies in visual grounding of language. Our work aims to bring important challenges actively studied in the dialog community to the multimodal setting. Specifically, we study the multimodal extension of the traditional dialog state tracking (DST) and the assistant API prediction tasks, which have been the key focus of dialog literature (Wu et al., 2019; Gao et al., 2019; Chao and Lane, 2019).

Compared to the conventional task-oriented dialog datasets (*e.g.*, MultiWoZ (Budzianowski et al., 2018)), the agent actions in SIMMC span across a diverse multimodal action space (*e.g.*, ROTATE, SEARCH, ADD_TO_CART). Our study thus shifts the focus of the visual dialog research from the token-level grounding of visual scenes to the task-level understanding of dialogs given multimodal context.

Semantic annotations for multimodal dialogs. Finally, we present a novel flexible schema for *seman-*

Dataset	Modality	Task	Provided Context		Updated	Annotation
			Q'er	A'er	Context	Granularity
Visual Dialog (Das et al., 2017)	Image	Q&A	N/A	Visual	N/A	N/A
CLEVR-Dialog (Kottur et al., 2019)	Simulated	Q&A	N/A	Visual	N/A	N/A
GuessWhat (De Vries et al., 2017)	Image	Q&A	N/A	Visual	N/A	N/A
Audio Visual Scene-Aware Dialog (Hori et al., 2018)	Video	Q&A	N/A	Visual	N/A	N/A
TalkTheWalk (de Vries et al., 2018)	Image	Navigation	Visual	Visual + Meta	Location	U ↔ A
Visual-Dialog Navigation (Thomason et al., 2019)	Simulated	Navigation	Visual	Visual + Meta	Location	U ↔ A
Relative Captioning (Guo et al., 2018)	Image	Image Retrieval	Visual	Visual + Meta	New Image	U ↔ A
MMD (Saha et al., 2018)	Image	Image Retrieval	Visual	Visual + Meta	New Image	U ↔ A
SIMMC (proposed)	Image/VR	Task-oriented	Visual	Visual + Meta	Situated	U ↔ A + Semantic

Table 1: **Comparison with the existing multimodal dialog corpora.** **Notations:** (U ↔ A) Utterance to action pair labels. (Task-oriented) Includes API action prediction, Q&A, recommendation, item / image retrieval and interaction. (Semantic) Dialog annotations such as NLU, NLG, DST, and Coref. (Situated) VR environment and/or new highlighted images.

tic annotations that we developed specifically for the natural multimodal conversations. The proposed SIMMC annotation schema allows for a more systematic and structural approach for visual grounding of conversations, which is essential for solving this challenging problem in the real-world scenarios. To the best of our knowledge, our dataset is the *first* among the related multimodal dialog corpora to provide fine-grained semantic annotations.

3 SIMMC Datasets

For SIMMC datasets, we focused on the shopping domain as it often induces rich multimodal interactions around browsing visually grounded items. As shown in Fig. 1, the setup consists of two human workers, a user and an assistant, conversing around a shopping scenario. The goal of the user is to interactively browse through an inventory of items while that of the assistant is to facilitate this conversation. In addition to having an interactive dialog, the assistant manipulates the co-observed environment to show off items from the shopping inventory. A conversational assistant model for the SIMMC datasets would need to (i) understand the user’s utterance

using both the dialog history and the state of the environment – the latter provided as multimodal context, and (ii) produce a multimodal response to the user utterance, including updates to the co-observed environment to convey meaningful information as part of the user’s shopping experience. We provide two SIMMC datasets with slightly different setups and modalities. See Tab. 2 for overall statistics.

The **SIMMC-Furniture (VR) Dataset** captures a scenario where a user is interacting with an assistant whilst browsing for furniture, *e.g.*, couch, or side table. Grounded in a VR environment (Unity Technologies, 2019) the assistant can manipulate items in the scene while engaging in conversation. We seed the conversation by presenting the user with either a high-level directive such as ‘*Shop for a table*’ or an image of a furniture item to shop for. The user is then connected randomly with a human assistant. The assistant can filter the catalog by attributes such as furniture category, price, color, and material, navigate through the filtered results and share their view with the user. As part of the dialog, the user can request to look closer at one of the options, or see other options. In response, the assistant can either zoom into an item, present an alternate view by rotating it, or look at the catalog description to answer further questions. To enable this, the environment is designed to transition between two states: (a) *Carousel*, which displays three filtered furniture items (top view, Fig. 1); and (b) *Focused*, which provides a zoomed in view of one item from the *carousel* view (bottom view, Fig. 1). The conversation continues for 6–12 turns until the user considers that they have reached a successful outcome. Tab. 10 shows example dialogs.

Statistics	Furniture (VR)		Fashion (Image)
	Text	Audio [†]	
Total # dialogs	6.4k	1.3k	6.6k
Total # utterances	97.6k	15.8k	71.2k
Avg # rounds / dialog	7.62	7.16	5.39
Avg # tokens (user)	11.0	N/A	11.10
Avg # tokens (assistant)	12.2	N/A	10.87

Table 2: **SIMMC Datasets Statistics.** [†]We also collected additional dialogs in aural medium where annotators exchanged audio messages instead of text.

The **SIMMC-Fashion (Image) Dataset** represents user interactions with an assistant to obtain recommendations for clothing items, *e.g.*, jacket, dress. Conversations are grounded in real-world images that simulate a shopping scene from a user’s point-of-view (POV). At the start of each dialog the user is presented with a randomly selected ‘seed’ image from the catalog to emulate (visually) that they are in the middle of shopping, as well as a sequence of synthetic *memories* of ‘previously viewed items’. In addition to the user’s context, the assistant has access to a broader catalog that allows for information lookup and item recommendation. We ask the user to browse and explore options by asking the assistant for recommendations based on the shared attributes, preferences, as referred from visual scenes, memories, and assistant-recommend items. The conversation continues for 6–10 turns until the user is assumed to be given a successful recommendation. Please refer to Tab. 11 for example dialogs.

For both datasets, the ground-truth of which items appear in each view is logged and included in the multimodal context. This allows the problem of computer vision to be sidestepped and focus on semantically combining the modalities. The datasets were collected through the SIMMC Platform (Crook et al., 2019), an extension to ParlAI (Miller et al., 2017) for multimodal conversational data collection and system evaluation. Note that even though we focus on English in this work, our data collection framework is language-agnostic and can be easily extended to other languages.

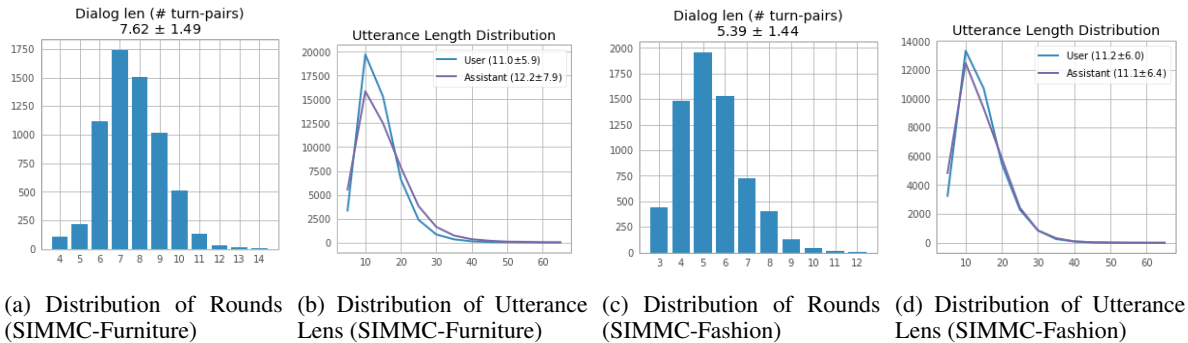


Figure 2: **SIMMC Datasets Analysis.** Distribution of Rounds and Utterance Lengths (# of tokens).

Dataset Analysis. SIMMC-Furniture has 6.4k dialogs with an average of 7.62 rounds (or turn pairs) leading to a total of about 97.6k utterances. Similarly, SIMMC-Fashion consists of 6.6k dialogs, each around 5.42 rounds on average, totaling 71.2k utterances. In addition to these sets, we also collect a smaller, audio-based SIMMC-Furniture dataset (1.3k dialogs) where the dialog exchanges are aural as opposed to written text.

In Fig. 2, we visualize: (a) *Distribution of rounds.* Dialogs in SIMMC-Furniture range from 4 (shorter ones are omitted from the dataset) to a maximum of 14 rounds, with 68% of the dialogs containing 7–9 rounds (Fig. 2a). Dialogs in SIMMC-Fashion range from 3–12 rounds at an average of 5.4 ± 1.4 rounds per dialog, as shown in Fig. 2c. We hope that this widespread range will help train models that can handle diverse conversations of varied lengths. (b) *Distribution of utterance lengths.* For both user and assistant, we tokenize their utterances and plot the distribution in Fig. 2b. For SIMMC-Furniture, the assistant utterances are slightly longer with higher variance at 12.2 ± 7.9 when compared to those from the user, at 11.0 ± 5.9 . A potential reason is that because the assistant has access to the catalog, it is expected to be more verbose while responding to description related queries (*User: Tell me more about the brown table*). However, we do not observe a similar trend for SIMMC-Fashion where user and assistant turns average around 11.2 ± 6.0 tokens per utterance (Fig. 2d). (c) *Catalog coverage.* Recall that both SIMMC datasets contain conversations in a shopping scenario grounded in a catalog of furniture and fashion items respectively. SIMMC-Furniture builds on a catalog of 179 items, where each dialog contains around 3.3 shares of different views between the user and assistant, and each furniture item is shared in roughly 45 dialogs. Similarly, SIMMC-Fashion contains 2098 items that appear in 32 dialogs on average, thus providing a rich catalog context to support interesting multimodal dialogs.

4 SIMMC Dialog Annotations

Building a task-oriented multimodal conversational model introduces many new challenges, as it requires both action and item-level understanding of multimodal interactions. While most of the previous multimodal datasets provide surface-level annotations (*e.g.*, utterance to multimodal action pairs), we believe it is critical to provide the semantic-level fine-grained annotations that ground the visual context, allowing for a more systematic and structural study for visual grounding of conversations. Towards this end, we develop a novel *SIMMC ontology* that captures the detailed multimodal interactions within dialog flows. Note that these dialog annotations are collected after the dialog data collection stage, with the help of professional linguists. In this section, we describe the proposed SIMMC ontology and the hierarchical labeling language centered around *objects* (Sec. 4.1 and 4.2), and the multimodal coreference schema that links the annotated language with the co-observed multimodal context (Sec. 4.3).

4.1 SIMMC Annotation Ontology

The SIMMC ontology provides common semantics for both the assistant and user utterances. The ontology is developed in the Resource Development Framework (RDF) and is an expansion of the Basic Formal Ontology (Arp et al., 2015). It consists of four primary components:

- **Objects:** A hierarchy of objects is defined in the ontology. This hierarchy is a rooted tree, with finer-grained objects at deeper levels. Sub-types are related to super-types via the *isA* relationship, *e.g.*, SOFA *isA* FURNITURE. Fine-grained objects include USER, DRESS, and SOFA.
- **Activities:** A hierarchy of activities are defined as a sub-graph of objects within the ontology. These represent activities the virtual assistant can take like GET, REFINE, and ADD_TO_CART.
- **Attributes:** A given object has a list of attributes which relate that object to other objects, to primitive data types, or to enums. Finer-grained objects inherit the attributes of their parents. There are restrictions on the available types for both the domain and range of attributes. For example, a SOFA can be related to a COMPANY via the *brand* attribute. A PERSON can be related to an item of CLOTHING via the *attentionOn* attribute. The *takesArgument* attribute relates Activities and the objects they act upon.
- **Dialog Acts:** A hierarchy of dialog acts is also defined as a sub-graph of objects within the ontology. Dialog acts indicate the linguistically motivated purpose of the user or system’s utterance. They define the manner in which the system conveys information to the user and vice versa. Examples of dialog acts include: ASK, INFORM, and PROMPT. Dialog acts are related to the activities that they act upon via the *takesArgument* attribute. Tab. 9 lists the activities and dialog acts used in our work.

4.2 SIMMC Labeling Language

From the SIMMC ontology, we derive a compositional, linearized, and interpretable labeling language for linguistic annotation, allowing for the representation of the natural language utterances as well-formed subgraphs of the ontology (Kollar et al., 2018). The labeling language consists of intents and slots (Gupta et al., 2006). Intents are taken to represent instances of the types they are composed of and take one of two forms: 1) DIALOG_ACT:ACTIVITY:OBJECT or 2) DIALOG_ACT:ACTIVITY:OBJECT.attribute. Only combinations of objects and attributes declared to be valid in the ontology are made available in the labeling language. Within these intents, slots further specify values for attributes of objects, activities, and attribute types. In the basic case, slots take the form of attributes of the intent-level objects and restrict those attributes. More complex cases include slot-in-slot nesting to restrict the type of the embedding slot, object-attribute combinations for type-shifting contexts, *i.e.*, utterances in which an intent-level object is identical to the range of another object’s property, and a system of indexing to restrict objects introduced within the intent. Crucially, the labeling language is speaker agnostic. It makes no distinction in the parses of the user’s utterance versus those of the assistant.

A number of additional conventions are placed on the annotation task to ensure consistency and accuracy, which are detailed in Appendix B. See Tab. 10 and Tab. 11 in Appendix G for annotated dialog examples that show our SIMMC ontology in action for both our datasets.

4.3 SIMMC Coreference Annotations

Note that the proposed labeling language allows for the annotation of object types in a dialog, which may in turn refer to specific canonical listings from the underlying multimodal contexts. For example, given an annotated utterance “[DA:REQUEST:GET:CHAIR *Show me the back of it*]”, the annotated object ‘CHAIR’ (*it*) would refer to a specific catalog item, represented as a item id within the image metadata. To allow for structural grounding between the verbal and visual modalities in a shared catalog, we further annotate the mapping of object type mentions in the annotated utterance to the corresponding item id in the image metadata. The final SIMMC annotations thus capture the semantic relations of objects in multimodal contexts with their corresponding dialog annotations (activities, attributes and dialog acts), as outlined in the proposed SIMMC ontology (Sec. 4.1). We provide the detailed analysis of the datasets and the annotations in Appendix A.

5 SIMMC Tasks & Metrics

We define several offline evaluation tasks within the SIMMC framework to train conversational models on these new datasets using the fine-grained annotations that are provided. We first provide the general offline evaluation framework for defining SIMMC tasks (Sec. 5.1), and then present three major tasks that we focus on in this paper. These are primarily aimed at replicating human-assistant actions in order to enable rich and interactive shopping scenarios (Sec. 5.2).

5.1 Offline Evaluation Framework

Consider a generic SIMMC dialog $\mathcal{D} = \{(U_i, A_i, M_i, a_i)\}_{i=1}^{N_r}$ that is N_r rounds long, where U_i and A_i are the user and assistant utterances, M_i is the domain-specific multimodal context, and a_i is the action (API call) taken by the assistant at round i , respectively. Formally, a task is defined as: At each round t , given the current user utterance U_t , the dialog history $H_t = (U_i, A_i)_{i=1}^{t-1}$, multimodal context M_t , predict the assistant action a_t along with the free-form, natural language assistant response A_t .

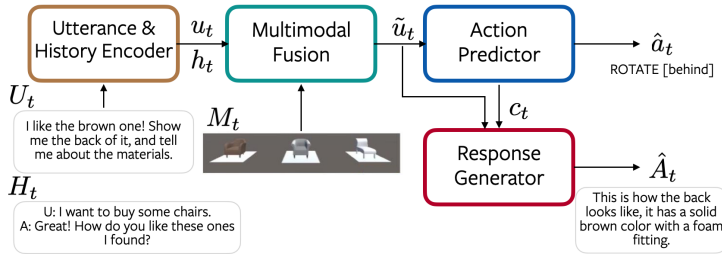


Figure 3: Assistant Model Architecture Overview (Sec. 6): utterance and history encoder, multimodal fusion, action predictor, and response decoder. Example taken from Fig. 1.

Model	Functions
HAE	$u_t = \text{LSTM}(U_t); h_t = \emptyset$
HRE	$u_t = \text{LSTM}(U_t)$ $\tilde{h}_t^{(i)} = \text{LSTM}([U_i, A_i])$ $h_t = \text{Attention}(u_t, [\tilde{h}_t^{(i)}]_{i=1}^{t-1})$
MN	$u_t = \text{LSTM}(U_t)$ $h_t = \text{LSTM}([u_i]_{i=1}^{t-1})$
T-HAE	$u_t = \text{Transformer}(U_t); h_t = \emptyset$

Table 3: Overview of SIMMC models. See Sec. 6 for details.

5.2 SIMMC Tasks

The proposed offline evaluation framework has a three-fold advantage: (a) It accurately represents the scenario encountered by a SIMMC model during deployment. In other words, models trained for the above task can be deployed to interact with humans to provide a situated, interactive, multimodal conversation. (b) Instead of evaluating the performance on the entire dialog, we evaluate models on a per-turn basis with the ground-truth history. This avoids taking the conversation out of the dataset and reduces the dependency on a user simulator, with the caveat of not encouraging the model to be able to learn multiple equally valid routes to satisfy the user’s request. (c) Finally, it facilitates us to define and evaluate several sub-tasks such as action prediction, response generation, and dialog state tracking, within SIMMC, which allows us to bootstrap from prior work on these sub-tasks.

Task 1: Structural API Call Prediction. This task involves predicting the assistant action a_t as an API call along with the necessary arguments, using H_t, M_t, U_t as inputs. For example, enquiring about an attribute value (e.g., price) for a shared furniture item is realized through a call to the *SpecifyInfo*

API with the *price* argument. A comprehensive set of APIs for our SIMMC dataset is given in Tab. 8. Apart from these APIs, we also include a *None* API call to catch situations without an underlying API call, e.g., a respond to ‘U: Can I see some tables?’ as ‘A: What color are you looking for?’ does not require any API calls. Action prediction is cast as a round-wise, multiclass classification problem over the set of APIs, measured using $1 - 0$ accuracy of predicting the action taken by the assistant during data collection. However, we note that there could be several actions that are equally valid in a given context. For instance, in response to ‘U: Show me some black couches.’, one could show black couches ‘A: Here are a few.’ or enquire further about specific preferences ‘A: What price range would you like to look at?’. Since accuracy does not account for the existence of multiple valid actions, we use perplexity (defined as the exponential of the mean log-likelihood) alongside accuracy. To also measure the correctness of the predicted action (API) arguments, we use attribute accuracy compared to the collected datasets.

Task 2: Response Generation. This task measures the relevance of the assistant response A_t in the current turn. We evaluate in two ways, as a: (a) Conditional language modeling problem, where the closeness between the generated and ground-truth response is measured through using BLEU-4 score (Papineni et al., 2002), and, (b) Retrieval problem, where performance of the model to retrieve the ground-truth response from a pool of 100 candidates (randomly chosen unique to each turn) is measured using standard retrieval metrics like recall@k ($k = 1, 5, 10$), mean rank, and mean reciprocal rank.

Task 3: Dialog State Tracking (DST). The dialog annotations collected using the flexible ontology enable us to study dialog state tracking (DST) in SIMMC, aside from providing additional supervision to train goal-driven agents. As mentioned in Sec. 4, the user and assistant utterances are accompanied with a hierarchy of *dialog act* labels and text spans for the corresponding slots or attributes, if any. The goal of DST is to systematically track the dialog acts and the associated slot pairs across multiple turns. We use the intent and slot prediction metrics (F1), following prior work in DST (Henderson et al., 2014).

6 Modeling for SIMMC Tasks

We now propose several models building on top of prior work and train them on the tasks formulated in Sec. 5 to benchmark the SIMMC dataset. We define two classes of models for the SIMMC tasks: (1) Assistant models, which aims at mimicking the assistant actions and responses (Task 1 & 2), and (2) User belief tracking model (Task 3) that output semantic parses of user utterances, agnostic of future assistant actions. Our principal Assistant model architecture is illustrated in Fig. 3, which is composed of four main components: Utterance and History Encoder, MultiModal Fusion, Action Predictor, and Response Generator. On the other hand, our user belief model builds upon the state-of-the-art DST models and extend them to accommodate for multimodal input. Inspired by (Hosseini-Asl et al., 2020), we adapt one such model and finetune a pretrained GPT-2 language model (Radford et al., 2019) to both action prediction and belief tracking.

Utterance & History Encoder. The utterance and history encoder takes as input the user utterance at the current round U_t and the dialog history so far H_t , to produce the utterance encoding u_t and history encoding h_t to capture the respective textual semantics. Inspired from prior work, we consider several utterance and history encoders, whose functional forms are outlined in Tab. 3. We embed each token in the input sequences (U_t or H_t) through learned word embeddings of size D_W , which are further fed into the encoders. These output $u_t \in \mathbb{R}^{N_U \times D_H}$ and $h_t \in \mathbb{R}^{t-1 \times D_H}$, where D_H is the embedding size, N_U is length of the user utterance. **(a) History-Agnostic Encoder (HAE)** ignores the dialog context H_t to only encode the user utterance through an LSTM (Hochreiter and Schmidhuber, 1997) for the downstream components. **(b) Hierarchical Recurrent Encoder (HRE)**(Serban et al., 2016) models dialogs at two hierarchical recurrence levels of utterance and turn. The utterance encoder LSTM operates at the former, while a history LSTM that consumes the hidden states of utterance encoder LSTM from all the previous rounds ($[u_i]_{i=1}^{t-1}$) operates at the latter. **(c) Memory Network (MN) encoder** (Sukhbaatar et al., 2015) treats dialog history H_t as a collection of memory units comprising user and assistant utterance pairs concatenated together, and uses the current utterance encoding u_t to selectively attend to these units to produce the utterance-conditioned history encoding h_t . **(d) Transformer-based History-Agnostic Encoder (T-HAE)** is a variant of HAE with the LSTMs replaced with Transformer units (Vaswani et al.,

2017) that achieved state-of-the-art results in language modeling (Devlin et al., 2019).

Multimodal Fusion. This component fuses semantic information from the text (u_t and h_t) and the multimodal context M_t (described in Sec. 7), to create the fused context tensor $\tilde{u}_t \in \mathbb{R}^{N_U \times 2D_H}$, which is double the size of u_t in the last dimension. In our setup, the multimodal context is modelled as a tensor of size $M_t \in \mathbb{R}^{N_M \times D_M}$, where N_M is the number of multimodal units for the current round t and D_M is the multimodal embedding size. Note that all of our models have the same architecture to fuse multimodal information. At a high level, we first embed M_t to match its size to D_H using a linear layer followed by a non-linearity (ReLU) (Eq. 1), then use the utterance encoding u_t to attend to the multimodal units (Eq. 2), and finally fuse the attended multimodal information with $\tilde{u}_t = [u_t; m_t]$. More concretely,

$$\tilde{M}_t = \text{Tanh}(\text{Linear}(M_t)), \quad (1) \quad m_t = \text{Attention}(u_t, \tilde{M}_t, \tilde{M}_t), \quad (2)$$

where Attention operator for a query Q over the key K (of size D_K) and value V is defined as

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_K}}\right)V. \quad (3)$$

Action Predictor. Using the fused context \tilde{u}_t , the Action Predictor predicts the appropriate action (API) \hat{a}_t and the corresponding API arguments to be taken by the assistant. The former is a multi-class classification that chooses from a set of actions (APIs) while the latter is a multi-way classification modelled as a set of binary classifiers one for each attribute like *category*, *color*, *price*, etc. For a list of APIs and their arguments supported in our work, see Tab. 8 in Appendix D. First, the tensor \tilde{u}_t is transformed into a vector $q_t \in \mathbb{R}^{N_H}$ through self-attention through the attention parameter θ_{AP} (Eq. 4). Next, we learn a classifier (MLP) that takes in q_t to predict the distribution over the possible APIs (Eq. 5). In addition, we also learn several binary classifiers (MLP) one each for the corresponding API arguments. Having predicted the structured API calls, we execute them and encode the output as action context $c_t \in \mathbb{R}^{N_A \times D_A}$, where N_A is the number of context units and D_A is action context embedding size. The dataset-dependent specifics about the API call output encoding c_t are in Sec. 7. Finally, c_t and \tilde{u}_t feed into the last component to generate the assistant response. As the training objective, we minimize the cross entropy loss \mathcal{L}_a for both the action and action attributes.

$$q_t = \text{Attention}(\theta_{AP}, \tilde{u}_t, \tilde{u}_t). \quad (4) \quad p(\hat{a}_t|U_t, M_t) = \text{Softmax}(\text{Linear}(q_t)). \quad (5)$$

Response Generator. As the last component, the response generator (decoder) generates the assistant response \hat{A}_t . In our work, we model it as a language model conditioned on both c_t and \tilde{u}_t . The former ensures that the response is influenced by the API call output while the latter maintains semantic relevance to the user utterance. For example, the response to ‘*Show me black couches less than \$500*’ depends on the availability of such couches in the inventory and could lead to either ‘*Here are some*’ or ‘*Sorry, we do not have any black couches cheaper than \$500*’. For models that use LSTM for user and history encoders, the response decoder is also an LSTM with attention over fused context \tilde{u}_t and action API output c_t at every decoding time step, similar to (Bahdanau et al., 2014). Similarly, we use a Transformer-based decoder for the other models to ensure consistent underlying architecture (either LSTM or transformer). Like any conditional language model, we decode individual tokens at each time step to generate \hat{A}_t , and minimize the negative loglikelihood \mathcal{L}_A of the human response under the model during training.

Dialog State Tracking (DST). In contrast to the assistant model that mimics the assistant actions and responses (Task 1 & 2), the user belief model aims to output semantic parses of user-side dialog (b_t), strictly given the multimodal contexts available to the user, and agnostic of future assistant actions. Thus, $b_t = \text{DST}(U_t, H_t, M_t)$. We utilize the state-of-the-art DST models from the recent literature: TRADE (Wu et al., 2019), which implements a pointer network that generates text spans for each slot, and an approach similar to SimpleTOD (Hosseini-Asl et al., 2020), which fine-tunes the pre-trained GPT-2 language model to output the user belief state labels (Radford et al., 2019).

In addition, we extend the SimpleTOD model to allow for multimodal input (SimpleTOD+MM). Specifically, we cast the belief tracking problem as a causal language modeling task, where belief labels and multimodal contexts are represented as additional tokens. A single training sequence can then

be represented as the concatenation of input and target output $y_t = [H_t; M_t; U_t; b_t]$, where both M_t and b_t are cast as string tokens of key value pairs. The language model is then fine-tuned to learn the joint probability with $p(x_t) = \prod_{i=1}^n p(x_{t,i}|x_{t,<i})$ for all n tokens in a sequence. At inference time, we provide the user input context $x_t = [H_t; M_t; U_t]$ as a seed for the language model, and parse the generated output to obtain the structural representation of user belief states.

We further extend the SimpleTOD+MM model to Tasks (1) and (2) by adding actions and assistant responses to the concatenation of input and target output, *i.e.*, $[H_t; M_t; U_t; b_t; a_t; A_t]$. At test time, we provide the input context plus oracle belief state $[H_t; M_t; U_t; b_t]$ and parse the generated response to extract the action and system utterance. We refer to this model as STOD++.

7 Experiments & Results

Dataset Splits and Baselines. Our models are learned on randomly sampled `train` (60%), model hyperparameters chosen via early stopping using performance on `dev` (10%), and evaluation numbers reported on the unseen `testdev` (15%). In addition to the models described in Sec. 6, we consider two simple baselines that use TF-IDF features for utterance and history encoders for action prediction, and LSTM-based language model (LSTM) trained solely on assistant responses, and compare against them.

Dataset-specific Model Details. We provide details around modeling multimodal context M_t and encoding action (API call) output c_t for each of the SIMMC datasets below.

A. SIMMC-Furniture (VR). Since the data collection for SIMMC-Furniture is grounded in a co-observed virtual 3D environment (Sec. 3), its state becomes the multimodal context M_t . For both *carousel* and *focused* environment states, we concatenate the furniture item representation in the corresponding slot (or zero vector if empty) with its positional embedding (*‘left’, ‘center’, ‘right’, ‘focused’*) that are jointly learned, to give $M_t \in \mathbb{R}^{N_M \times D_M}$ with $N_M = 3$ (carousel) or $N_M = 1$ (focused). In addition, each furniture item is represented with the concatenated GloVe embeddings (Pennington et al., 2014) of its attributes like category, color, intended room, *etc.* Similarly, we construct the action output $c_t \in \mathbb{R}^{N_A \times D_A}$ using the environment representation after executing the necessary structural API call, *e.g.*, *search* for an item or *focus* on an existing item. The information seeking action *SpecifyInfo* is an exception, for which c_t is the GloVe embedding of the attributes of the desired item.

B. SIMMC-Fashion (Image). Dialogs in SIMMC-Fashion use a fashion item (updated as the conversation progresses) and a sequence of ‘previously viewed items’ (memory) as context (Sec. 3). To reflect this scenario, we extract the representations for each fashion item using concatenated GloVe embeddings of its attributes (similar to SIMMC-Furniture) in addition to learning the source embedding (*‘memory’* or *‘current’* item), as the multimodal context $M_t \in \mathbb{R}^{4 \times D_M}$. Akin to SIMMC-Furniture, c_t is modeled simply as the updated multimodal state M_t after executing the current API.

Supervision. We learn SIMMC models end-to-end by jointly minimizing the sum of the action prediction and the response generation losses, *i.e.*, $\mathcal{L}_a + \mathcal{L}_A$. To extract supervision for API call prediction (along with attributes), we utilize both the assistant (Wizard) interface activity during data collection (Sec. 3) and the fine-grained NLU annotations. Our implementation details are in Appendix E.

Results. Tab. 4 summarizes the performance of SIMMC Assistant models on structural API prediction and response generation.

The key observations are: (a) All SIMMC neural models (HAE, HRE, MN, T-HAE) outperform the baselines (TF-IDF and LSTM) across all metrics for both the datasets. (b) HRE consistently achieves the highest API prediction accuracy for SIMMC-Furniture (80.0%, jointly with HAE) and SIMMC-Fashion (81.9%, jointly with HAE and MN). STOD++ achieves 61.4% accuracy on attributes, an overwhelming 7% point improvement over T-HAE for SIMMC-Furniture, benefiting from having access to the oracle belief state where the user requested attributes are formally represented. (c) For response generation, HRE has superior BLEU score for SIMMC-Furniture and HRE for SIMMC-Fashion. Surprisingly, T-HAE has the least BLEU scores amongst SIMMC models perhaps due to resorting to safe, frequent

Model	Task 1. API Prediction			Task 2. Response Generation					
	Acc \uparrow	Perp \downarrow	A.Acc \uparrow	BLEU \uparrow	r@1 \uparrow	r@5 \uparrow	r@10 \uparrow	Mean \downarrow	MRR \uparrow
SIMMC-Furniture									
TF-IDF	77.1	2.59	57.5	-	-	-	-	-	-
LSTM	-	-	-	0.022	4.1	11.1	17.3	46.4	0.094
HAE	79.7	1.70	53.6	0.075	12.9	28.9	38.4	31.0	0.218
HRE	80.0	1.66	54.7	0.075	13.8	30.5	40.2	30.0	0.229
MN	79.2	1.71	53.3	0.084	15.3	31.8	42.2	29.1	0.244
T-HAE	78.4	1.83	53.6	0.044	8.5	20.3	28.9	37.9	0.156
STOD++ \dagger	72.2	-	61.4	0.155	-	-	-	-	-
SIMMC-Fashion									
TD-IDF	78.1	3.51	57.9	-	-	-	-	-	-
LSTM	-	-	-	0.022	5.3	11.4	16.5	46.9	0.102
HAE	81.0	1.75	60.2	0.059	10.5	25.3	34.1	33.5	0.190
HRE	81.9	1.76	62.1	0.079	16.3	33.1	41.7	27.4	0.253
MN	81.6	1.74	61.6	0.065	16.1	31.0	39.4	29.3	0.245
T-HAE	81.4	1.78	62.1	0.051	10.3	23.2	31.1	37.1	0.178

Table 4: Results for: (1) **API prediction** via accuracy, perplexity and attribute accuracy, and, (2) **Response generation** via BLEU, recall@k ($k=1,5,10$), mean rank, and mean reciprocal rank (MRR). Std Errors: $< 0.5\%$ for Acc, A.Acc, r@1, r@5, r@10, mean; 0.005 for BLEU and MRR. \dagger Uses oracle belief state.

Model	T3. DST	
	In.F1 \uparrow	Sl.F1 \uparrow
SIMMC-Furniture		
TRADE	-	45.5
SimpleTOD	75.0	50.1
SimpleTOD+MM	74.1	60.2
SIMMC-Fashion		
TRADE	-	32.8
SimpleTOD	56.5	37.3
SimpleTOD+MM	59.1	43.5

Table 5: Results for: (3) **Dialog State Tracking (DST)**, measured with Intent and Slot prediction F1 metrics. \uparrow : higher is better, \downarrow : lower is better. Bold denotes the best for each metric.

responses. (d) The confusion matrix for HRE on SIMMC-Furniture (Appendix F) reveals a high confusion between *SearchFurniture* and *None*.

This is intuitive as searching for an item or further obtaining user preferences to narrow the search are equally valid actions for their context. Note that the proposed assistant models do not leverage the rich, fine-grained annotations of the SIMMC datasets (understandably so) as they are adaptations of existing state-of-the-art models.

Tab. 5 presents the performance of the state-of-the-art DST models on the SIMMC datasets. It can be seen that the pretrained GPT-2 based SimpleTOD models outperform the TRADE baseline. Note that the original TRADE implementation does not include the dialog act prediction, hence it is not reported here as well. When the multimodal contexts are added as input (SimpleTOD+MM), the performance improves upon the text-only SOTA model (SimpleTOD) on both datasets, especially in the slot prediction metrics. This demonstrates the efficacy of grounding the multimodal contexts for DST, by better resolving the multimodal coreferences. In general, the performance on the SIMMC-Fashion dataset is typically better than on the SIMMC-Furniture dataset. This could be due to the nature of the dialogs in the SIMMC-Fashion dataset, which involves more natural and casual utterances, as evident in the lower annotator agreement as well (Appendix C).

8 Conclusion

In this work, we presented **Situated Interactive Multi-Modal Conversations (SIMMC)**, an important new direction towards building next generation virtual assistants with evolving multimodal inputs. In particular, we collected two new datasets using the SIMMC platform, and provided the contextual NLU and coreference annotations on these datasets, creating a new SIMMC task for the community to study. We established several strong baselines for some of the tasks enabled by the datasets, showcasing various uses of the datasets in real-world applications. The fine-grained annotations we collected open the door for studying several different tasks in addition to the ones highlighted in this work, which we leave as future work for the community to tackle.

Acknowledgements

We thank Pararth Shah, Oksana Buniak, Semir Shafi, Ümit Atlamaz, Jefferson Barlew, Becka Silvert, Kent Jiang, Himanshu Awasthi, and Nicholas Flores for their invaluable technical contributions to the data collection platforms, annotation schema development, annotation process, tooling and coordination. We also extend many thanks to all the annotators who meticulously labelled these datasets.

References

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *ICCV*.
- Robert Arp, Barry Smith, and Andrew D. Spear. 2015. *Building Ontologies with Basic Formal Ontology*. MIT Press.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In *EMNLP*.
- Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. In *INTERSPEECH*.
- Meng Chen, Ruixue Liu, Lei Shen, Shaozu Yuan, Jingyan Zhou, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. The JDDC corpus: A large-scale multi-turn Chinese dialogue dataset for E-commerce customer service. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 459–466, Marseille, France, May. European Language Resources Association.
- Paul A. Crook, Shivani Poddar, Ankita De, Semir Shafi, David Whitney, Alborz Geramifard, and Rajen Subba. 2019. SIMMC: Situated Interactive Multi-Modal Conversational Data Collection And Evaluation Platform. *ASRU*.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *CVPR*.
- Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*.
- Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.
- Shuyang Gao, Sanchit Agarwal, Abhishek Sethi, and Tagyoung Chun, and Dilek Hakkani-Ture. 2019. Dialog state tracking: A neural reading comprehension approach. In *SIGDIAL*.
- Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogerio Feris. 2018. Dialog-based interactive image retrieval. In *NeurIPS*.
- N. Gupta, G. Tur, D. Hakkani-Tur, S. Bangalore, G. Riccardi, and M. Gilbert. 2006. The at t spoken language understanding system. *TASLP*.
- Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *SIGDIAL*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.
- Chiori Hori, Anoop Cherian, Tim K. Marks, and Florian Metze. 2018. Audio visual scene-aware dialog track in dstc8. *DSTC Track Proposal*.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.
- Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- Thomas Kollar, Danielle Berry, Lauren Stuart, Karolina Owczarzak, Tagyoung Chung, Lambert Mathias, Michael Kayser, Bradford Snow, and Spyros Matsoukas. 2018. The Alexa meaning representation language. In *NAACL*.

- Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2019. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*.
- A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. 2017. ParlAI: A Dialog Research Software Platform. *arXiv*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *AAAI*.
- Amrita Saha, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *AAAI*.
- Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.
- Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. *arXiv preprint arXiv:1907.04957*.
- Unity Technologies. 2019. Unity. <https://unity.com/>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *ACL*.

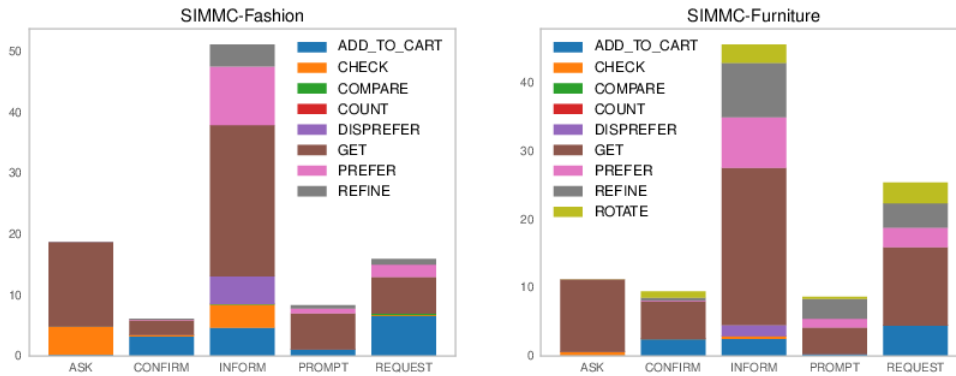


Figure 4: Distribution of Dialog Acts and Activities in the SIMMC datasets. See Sec. A for details.

A Dataset & Annotation Analysis

Annotation Analysis. Using the unified ontology framework described in Sec. 4.1, we annotate both the user and assistant utterances of the SIMMC datasets. There are effectively 5 dialog acts which are respectively combined with 9 activities for (SIMMC-Furniture) and 8 activities for (SIMMC-Fashion); the latter by design excludes COUNT and ROTATE. A detailed list with examples is in Appendix, Tab. 9. Not all combinations of dialog acts and activities are observed in our dataset, *i.e.*, about 38/45 for SIMMC-Furniture and 32/40 for SIMMC-Fashion respectively. For instance, a REQUEST:DISPREFER utterance is an invalid combination. The key takeaways from Fig. 4 are: (a) INFORM is the most dominant dialog act (50% in SIMMC-Fashion and 45% in SIMMC-Furniture). This is intuitive as conversations in shopping domain require the user to *inform* the assistant of their preferences, while the assistant *informs* the user about the item attributes and availability. (b) Interestingly, GET is the dominant activity across most dialog acts, where the assistant either *gets* new items or additional information about existing items that the user is perusing. (c) The relatively low occurrence of the CONFIRM dialog act perhaps arises from the effectiveness of the human assistant agent. This is desirable to avoid learning assistant models that excessively repeat user requests, *e.g.*, repeatedly seek explicit confirm, as this leads to lower user satisfaction. Note that this analysis of the dialog act and activity distribution is per sentence, with an utterance occasionally containing multiple sentences (see Fig. 1 for an example).

User Satisfaction Metrics. Since SIMMC datasets aim at goal-oriented dialog, we also collect turn-level and dialog-level user satisfaction scores in the range of 1-5 as part of the data collection. The dialog-level user satisfaction scores for the SIMMC-Furniture dataset average at 4.69 ± 0.77 , showing a heavy concentration around 5. Since the dialogs are collected between humans interacting with each other, we hypothesize that the the assistant (wizard) is able to efficiently respond to user requests, leading to high satisfaction scores. Similar trends were observed across different metrics for both datasets. Therefore, we drop further analysis on this front due to the absence of a clear signal in these collected metrics.

B Details for SIMMC Labeling Language

A number of additional conventions are placed on the annotation task to ensure consistency and accuracy.

Type ambiguity. When an object appears in an utterance, the most fine-grained type is annotated. For example, in the utterance “*Show me some dresses*”, the token ‘dresses’ needs to be annotated as DRESS, as opposed to a coarser-grained type CLOTHING. When more than one fine-grained type is possible, the annotator utilizes a parent-level coarse-grained type instead. Thus the assigned type is the finest-grained type that still captures the ambiguity.

Attribute ambiguity. Attributes are annotated when they are unambiguous. When there is uncertainty in the attribute that should be selected for the representation, the annotator falls back to a more generic attribute. For example when asking about an FURNITURE item the user may specify a particular dimen-

sion, e.g. *I want a couch that is 2 feet wide*. In this case, *2 feet* can be annotated with the specific attribute *width*. However, if the dimension is not specified, the more general attribute *dimensions* would be used.

Attribute inverses. When an attribute can be annotated in two different directions, a canonical attribute is defined in the ontology and used for all annotations. For example, *attentionOn* and *inAttentionOf* are inverses. The USER object is connected to FURNITURE or CLOTHING objects via *attentionOn* if the user is looking at an instance of these objects. Inversely, those same objects are connected to the USER object via the *inAttentionOf* attribute. The former is designated as the canonical attribute in this case, and used for labeling purposes.

Smart prefixes. Attribute slots are prefixed by *A* and *O* respectively to indicate whether they serve to restrict the intent-level Activity or Object. This is primarily for human-annotator convenience. For example the attribute *amount* is an attribute of the Activity GET. The attribute *color* is an attribute of CLOTHING. Annotating an assistant response like *'I found five green dress'* yields spanning *five* with *A.amount* and *green* with *O.color*.

Attribute variables. The attribute *.info* is employed when the speaker's intent targets more than one attribute simultaneously. The specific attributes being targeted are then identified with the INFO smart prefix. For example, *'What is the color and brand of this skirt?'* is annotated with the intent *DA:ASK:GET:SKIRT.info* and the tokens *color* and *brand* are labeled as *INFO.color* and *INFO.brand* respectively.

C Details for NLU/NLG/Coref Data Annotation

Data were annotated in two stages: (1) NLU/NLG followed by (2) image-based coreference annotations.

During the NLU/NLG stage, annotators were provided full dialog context for a single dialog and asked to annotate both the user and assistant's utterances. Image context was not available, and annotators were instructed to use dialog context only up to the target utterance. Essentially all (98.4%) annotations were single-annotated by annotators who passed an evaluation test; while 1.6% were double-annotated. In cases of disagreement between two annotators, a third annotator either selected one of the proposed annotations, or overrode both with a new one.

In order to estimate and improve quality, we double-annotated an additional 12.6% of the data after the fact and applied two measures of inter-annotator agreement: exact matches between semantic parses, and a modified F1 score. 50% of fashion and 60% of furniture annotations were exact matches. Given sample sizes, this corresponds respectively to 95% confidence intervals of 49-50% and 58-63%. In contrast to this binary exact measure, the F1 measures can assume values between 0 and 1. Using this measure, furniture annotations were 73.8% similar while fashion annotations were 72.5% similar.

During image-based coreference, annotators were provided all dialog and image context up until the turn in question. Review of the process suggested it was easy enough for the high-skilled pool annotators to perform without quality checks. By their own account, annotators self-reported 98% confidence in their decision to link an object to an intent; and 98% confidence in their specific choice of object given a link was required.

Tab. 6 presents the Object Classes that were made available to the annotators for annotation as well as the attributes of these Classes. Attributes are listed alphabetically and type information is provided. Note that for readability attributes derived via inheritance from supertype to subtype are not repeated. Classes that were exposed to annotators but had no attributes are not presented here. Attribute ambiguity is indicated by indenting.

Tab. 7 presents the Activity Classes that were made available to the annotators for annotation as well as the attributes of these Classes (see Tab. 9 for examples and definitions). Type information and attributes are provided. Note that for readability attributes derived via inheritance from supertype to subtype are not repeated. All Activities had the attributes *amount* an INTEGER, *endTime* and *startTime* (DATE_TIMES). Only Activities with additional attributes are listed below.

<u>CLOTHING</u>	ageRange, amountInStock, availableSizes, brand, clothingCategory, clothingStyle, color, condition, customerRating, embellishment, forGender, forOccasion, forSeason, itemDescription, madeIn, material, ordinal, pattern, price, sequential, size, items, soldBy, warmthRating, waterResistance
<u>COMPANY</u>	headquarteredIn, name*, ordinal, sequential
<u>DATE_TIME</u>	date, month, time, week, weekday, year
<u>DISPLAY</u>	displayPostion (displayFirst, displaySecond, displayThird)
<u>DRESS</u>	dressStyle, hemLength, hemStyle, necklineStyle, sleeveLength, sleeveStyle, waistStyle
<u>EVENT</u>	duration, elapsedTime, endTime, eventType, hasPart, name, remainingTime, startTime
<u>FURNITURE</u>	ageRange, amountInStock, assemblyRequired, brand, color, condition, currentLocation, customerRating, decorStyle, dimensions (width, depth, height) era, filling, finish, foldable, hasStorage, intendedRoom, isAdjustable, isAntique, isVintage, madeIn, material, name, ordinal, owner, pattern, price, sequential, soldBy, swivels, upholstery, weight, weightCapacity
<u>HOLIDAY</u>	duration, endTime, name, startTime
<u>JACKET</u>	hemLength, hemStyle, jacketStyle, necklineStyle, sleeveLength, sleeveStyle, waistStyle
<u>LOCATION</u>	city, continent, country, currentDate, currentTime, region, state
<u>SITUATION</u>	agent, situationLocation, situationTime, situationType, theme
<u>SIZE</u>	ageSize, alphabeticSize, numericSize, ordinal, sequential, sizeType
<u>SKIRT</u>	hemLength, hemStyle, skirtStyle, waistStyle
<u>SWEATER</u>	necklineStyle, sleeveLength, sleeveStyle, sweaterStyle, waistStyle
<u>USER</u>	attentionOn, name

Table 6: List of object attributes in the SIMMC ontology

<u>CHECK</u>	check (STRING)
<u>COMPARE</u>	comp (OBJECT)
<u>COUNT</u>	countFrom (THING), countTo (THING), countUnit (STRING)

Table 7: List of activity attributes in the SIMMC ontology

D API Call List

Tab. 8 shows the list of all APIs supported in our SIMMC datasets.

E Implementation Details

All our models are trained using PyTorch (Paszke et al., 2019). We consider words (after converting them to lowercase) that occur at least 5 times in the training set, to yield model dictionaries of size 2619 and 2032 for SIMMC-Furniture and SIMMC-Fashion, respectively. We learn $D_W = 256$ dimensional word embeddings for each of these words that are fed into utterance and history encoder. All the LSTMs (2 layers) and Transformers (4 layers, 4 heads each, with 2048 internal state) have a hidden state of size $D_H = 256$, in our experiments. We optimize the objective function using Adam (Kingma and Ba, 2015) with a learning rate of 10^{-4} and clip the gradients by value to be within $[-1.0, 1.0]$. The model hyperparameters are selected via early stopping on the development set.

F Model Visualizations

The action API confusion matrix for hierarchical recurrent encoder (HRE) model for the SIMMC-Furniture dataset is given in Fig. 5.

API Name	Arguments
SIMMC-Furniture	
<i>SearchFurniture:</i> Search items using the item attributes	Category, color, intended room, material, price range, <i>etc.</i>
<i>SpecifyInfo:</i> Get and specify information (attributes) about an item	Material, price range (min–max), customer rating, <i>etc.</i>
<i>FocusOnFurniture:</i> Focus on an item to enlarge (for a better view)	Position of argument item on the carousel (left, center, right)
<i>RotateFurniture:</i> Rotate a focused furniture item in the view	Rotational directions (left, right, up, down, front, back)
<i>NavigateCarousel:</i> Navigate the carousel to explore search results	Navigating directions (next and previous)
SIMMC-Fashion	
<i>SpecifyInfo:</i> Get and specify information (attributes) about an item	Brand, price, customer rating, available sizes, colors, <i>etc.</i>
<i>Search(Database Memory):</i> Select a relevant image from either the database or memory, and specify information	Brand, price, customer rating, available sizes, colors, <i>etc.</i>

Table 8: List of APIs supported in our SIMMC datasets with attributes. We also include *None* as an action when no API call is required and *AddToCart* to specify adding an item to cart for purchase.



Figure 5: Confusion matrix for hierarchical recurrent encoder (HRE) on SIMMC-Furniture.

G Dataset Examples

See Tab. 10 and Tab. 11 in Appendix G for annotated dialog examples that show our SIMMC ontology in action for both our datasets.

Dialog Acts		
Name	Description	Examples
ASK	Used when the main intention of the utterance is information seeking, i.e. a question.	[DA:ASK:GET:DRESS.price How much is the dress?] [DA:ASK:GET:TABLE.color What color is [USER.attentionOn that] table?]
CONFIRM	Used when the utterance is asking for or giving confirmation for something that has been said in an earlier turn.	[DA:CONFIRM:GET:DRESS.price One moment while I find the dress's price.] [DA:CONFIRM:GET:TABLE.color I'll get that table's exact color information from the catalog.]
INFORM	Used when the main intention of the utterance is information providing.	[DA:INFORM:GET:DRESS.price The dress costs [O.price \$99.99].] [DA:INFORM:GET:TABLE.color That table is [O.color hunter green].]
PROMPT	Used when the main intention of the utterance is to suggest an action or prompt the user to take an action.	[DA:PROMPT:PREFER:DRESS What do you think of the dress?] [DA:PROMPT:ADD_TO_CART:TABLE Would you like me to add the table to your shopping cart?]
REQUEST	Used when the utterance is a request for action.	[DA:REQUEST:ADD_TO_CART:DRESS I want to buy that dress!] [DA:REQUEST:ROTATE:TABLE Show me a [A.rotateTo:SIDE side] view first.]
Activities		
Name	Description	Examples
ADD_TO_CART	Indicates an intent to purchase.	[DA:REQUEST:ADD_TO_CART:DRESS Add the [O.color green] one to my cart.] [DA:INFORM:ADD_TO_CART:TABLE I've added the [O.price \$50] table for check out.]
CHECK	Requests a yes/no and alternative questions be answered about an items attribute value.	[DA:REQUEST:CHECK:DRESS.color Is the dress [.check green] or [.check blue] ?] [DA:INFORM:CHECK:TABLE.color Yes, the table is [.check blue] .]
COMPARE	Requests two (or more) items be compared along a stated attribute.	[DA:REQUEST:COMPARE:DRESS.price Is the [R1.color green] [A.comp:DRESS.1 one] more expensive than [2:USER.attentionOn this] [A.comp:DRESS.2 dress]?] [DA:INFORM:COMPARE:TABLE.width The [R1.color blue] [A.comp:TABLE.1 table] is wider.]
COUNT	Requests the number of items fitting a certain description be returned.	[DA:REQUEST:COUNT:DRESS How many [O.color green] ones do you have?] [DA:INFORM:COUNT:TABLE I found [A.amount 5] [O.color blue] tables.]
DISPREFER	Indicates dislike for an item or attribute of that item.	[DA:INFORM:DISPREFER:DRESS [USER.attentionOn That] dress is ugly!] [DA:INFORM:DISPREFER:TABLE.price I'm not a fan of the cost of the table.]
GET	Requests some type of item or attribute of an item be retrieved.	[DA:REQUEST:GET:DRESS I'd like to a buy a dress.] [DA:INFORM:GET:TABLE.brand This table is made by [O.brand [.name Wind & Wool]]]
PREFER	Indicates like for an item or attribute of that item.	[DA:INFORM:PREFER:DRESS [USER.attentionOn That] dress is beautiful!] [DA:INFORM:PREFER:TABLE.price Wow what a bargain for the table!]
REFINE	Indicates additional constraints to restrict a search.	[DA:REQUEST:REFINE:DRESS.color Only show me [O.color green] dresses] [DA:INFORM:REFINE:TABLE.price I've limited results to tables [O.price under \$1000].]
ROTATE	Requests an item (of furniture) be rotated to see another view.	[DA:REQUEST:ROTATE:TABLE Can you show me the [A.rotateTo:BACK back] of the table?] [DA:CONFIRM:ROTATE:TABLE Yes, I'll provide the [A.rotateTo:BACK back] view momentarily.]

Table 9: List of **Dialog Acts** and **Activities** used in the SIMMC Annotation Ontology (Sec. 4) along with examples from both SIMMC-Furniture and Fashion (where applicable). We use a compositional, linearized, and interpretable annotation ontology that is unified for both the user and assistant utterances.

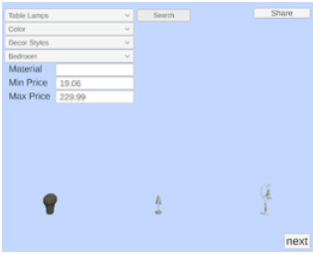
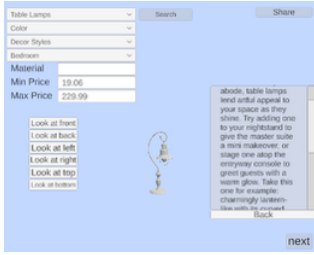
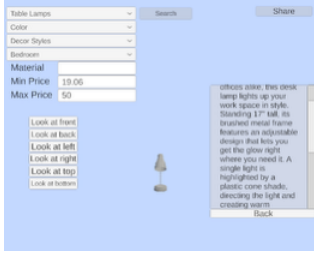
Situated Context	Dialog Utterances	Dialog Annotation
	U: I am looking for table lamps	[DA:REQUEST:GET:TABLE_LAMP I am look for table lamps]
	A: Are you looking for bedroom lamps?	[DA:PROMPT:REFINE:TABLE_LAMP .intendedRoom Are you looking for [.intendedRoom bedroom] lamps?]
	U: Yes, bedroom lamps	[DA:INFORM:REFINE:TABLE_LAMP .intendedRoom Yes, [.intendedRoom bedroom] lamps]
	A: If you're into a sophisticated style, this neutral gray finish lamp will fit any color palette. I'm showing you the Mentzer Table Lamp. It has a curved steel frame finished in chrome.	[DA:INFORM:GET:TABLE_LAMP If you're into a [O.decorStyle sophisticated] style, this [O.finish neutral gray] finish lamp will fit any color palette. I'm showing you the [O.name Mentzer] Table Lamp. It has a curved steel frame finished in [O.color chrome].]
	U: What is the cost Mentzer Table Lamp?	[DA:ASK:GET:TABLE_LAMP .price What is the cost [O.name Mentzer] Table Lamp?]
	A: This is priced at \$127.99	[DA:INFORM:GET:TABLE_LAMP .price This is priced at [O.price \$127.99]]
	U: Do you have anyone under \$50	[DA:REQUEST:GET:TABLE_LAMP Do you have anyone [O.price under \$50]]
	A: Yes I do! This Bergh 17 bedroom lamp is priced at \$38. It's 17 inches tall.	[DA:CONFIRM:GET:TABLE_LAMP Yes I do!] [DA:INFORM:GET:TABLE_LAMP This [O.name Bergh] 17 [O.intendedRoom bedroom] lamp is priced at [O.price \$38]. It's [O.height 17 inches] tall.]
	U: I'll take it!	[DA:REQUEST:ADD.TO.CART:TABLE_LAMP I'll take it!]

Table 10: Dataset Example: SIMMC-Furniture (VR)

Situated Context	Dialog Utterances	Dialog Annotation
	U: What can you tell me about the style of <u>that brown skirt</u> ?	[IN:ASK:GET:SKIRT.skirtStyle What can you tell me about the style of [USER.attentionOn this] skirt?]
	A: This style is loose ball gown.	[IN:INFORM:GET:SKIRT.skirtStyle This style is [O.skirtStyle loose ball gown]]
	U: Can you show me another brown skirt?	[IN:REQUEST:GET:SKIRT Can you show me [O.sequential another] [O.color brown]skirt?]
	A: Here's <u>another brown skirt</u> from Wind & Wool.	[IN:INFORM:GET:SKIRT Here's [O.sequential another] [O.color brown] skirt from [O.brand [.name Wind & Wool]].]
	U: How much for <u>this one</u> ?	[IN:ASK:GET:SKIRT.price How much for [USER.attentionOn this] one?]
	A: <u>This</u> costs \$139 and has a 3.86 rating.	[IN:INFORM:GET:SKIRT.info This costs [INFO.price \$139] and has a [INFO.customerRating 3.86] rating.]
	U: And how much is <u>the longer brown</u> I saw before?	[IN:ASK:GET:SKIRT.price And how much is the [O.hemLength longer] [O.color brown] I saw before?]
	A: <u>That one</u> is \$272.	[IN:INFORM:GET:SKIRT.price That one is [O.price \$272].]
	U: Put <u>the short brown one</u> in my cart.	[IN:REQUEST:ADD_TO_CART:SKIRT Put the [O.hemLength short] [O.color brown] one in my cart.]

Table 11: **Dataset Example: SIMMC-Fashion (Image)**. Multimodal coreferences are marked with underlines.