Learning to Embed Multi-Modal Contexts for Situated Conversational Agents

Anonymous NAACL submission

Abstract

The Situated Interactive Multi-Modal Conversations (SIMMC) 2.0 aims to create virtual shopping assistants that can accept complex multi-modal inputs, i.e. visual appearances of objects and user utterances. It consists of four subtasks, multi-modal disambiguation (MM-Disamb), multi-modal coreference resolution (MM-Coref), multi-modal dialog state tracking (MM-DST), and response retrieval and generation. While many task-oriented dialog systems usually tackle each subtask separately, we propose a jointly learned multi-modal encoder-decoder that incorporates visual inputs and performs all four subtasks at once for efficiency. This approach won the MM-Coref and response retrieval subtasks and nominated runner-up for the remaining subtasks using a single unified model at the 10th Dialog Systems Technology Challenge (DSTC10), setting a high bar for the novel task of multi-modal task-oriented dialog systems.\(^1\)

1 Introduction

A task-oriented dialog system aims to assist users accomplish certain tasks, such as executing actions or retrieving specific information, with natural language conversations. With the rising interest in multi-modal representation learning, the next generation of task-oriented virtual assistants is expected to handle conversations in such contexts, especially in the domain of vision-language (VL). For instance, a multi-modal dialog agent may help the user navigate a virtual clothing store and look for an object meeting the user’s criteria. In such cases, a successful dialog agent should be able to parse and understand multi-modal contexts.

To this end, SIMMC 2.0 (Kottur et al., 2021) proposes a situated multi-modal context in the form of co-observed, realistic scene set in virtual reality (VR) stores to incorporate the complexity of multi-modal task-oriented dialogs. The multi-modal subtasks, MM-Disamb and MM-Coref, intend to test the assistant’s capability to identify the need for disambiguating reference mentions and to ground them to the scene objects. While challenging, these are all essential to building a successful multi-modal dialog agent.

In this paper, we present our end-to-end, joint-learning approach to address this challenge in SIMMC 2.0. We adopt BART (Lewis et al., 2019) and attach task-specific heads so that the model can make predictions on all subtasks at once. To be more specific, our model performs MM-Disamb, MM-Coref, and response retrieval by the encoder and MM-DST and response generation in a string format by the decoder. We also integrate multi-modality into the model by extracting visual features of each object from a convolutional vision backbone then combining them with non-visual attributes. Our model is jointly trained on all subtasks and a couple of auxiliary objectives to help the model align the different modalities. For retrieval, we use in-batch negative samples for contrastive metric learning instead of creating a pool of separate training samples.

With modification for the competition setting, our model was ranked at the first place for MM-Coref and response retrieval with 75.8% coreference F1, 82.5% MRR, 72.5% R@1, 95.0% R@5, 98.4% R@10, and 1.9 mean rank in the official evaluation of DSTC10. Moreover, our model was nominated runner-up for all other subtasks, in which we achieved 93.8% disambiguation accuracy, 90.3% slot F1, 95.9% intent F1, and 0.295 BLEU-4. The results were obtained with only a single model and consistent with the results on the devset (i.e. validation) set, demonstrating a robust, common representation on all subtasks learned by the model.

---

\(^1\)Code is available at [https://anonymous.4open.science/r/NACCL_SUBMISSION_ANONYMOUS-1CD3](https://anonymous.4open.science/r/NACCL_SUBMISSION_ANONYMOUS-1CD3)
2 Related Work

Recent works on (uni-modal) task-oriented dialog systems remove the need for a pipeline composed of NLU (Liu and Lane, 2016), DST (Mrksic et al., 2017), POL (Wen et al., 2017), and NLG (Wen et al., 2015) modules leveraging pretrained language models (LM) that integrate all the modules in an end-to-end, auto-regressive manner (Ham et al., 2020; Hosseini-Asl et al., 2020; Yang et al., 2021). Given a dialog context, such systems sequentially generates belief state, system action, and response, making predictions based on decisions made by previous modules in the form of tokens and achieving superior results to the pipelined approaches. Some of these systems aim to learn the user preference from dialogs and recommend the object based on external knowledge base (KB) (Zhou et al., 2020).

In a similar context, building cross-modal models has recently gained attention in VL domain. Recent works develop VL models on top of transformer-based (Vaswani et al., 2017) pretrained LM and vision backbones, focusing on self-supervised pretraining methods to align joint embedding between different modalities. They achieve state-of-the-art performance in downstream tasks such as visual question answering (VQA), as shown in (Chen et al., 2020) and (Li et al., 2020). However, there are only a handful of works focusing on situated VL task-oriented dialog systems (Liao et al., 2018), where visual modality of the task is provided in a sanitized setting rather than a natural, situated scene.

3 SIMMC 2.0 Description

3.1 Dataset

SIMMC 2.0 (Kottur et al., 2021)\(^2\) follows the setting of SIMMC 1.0 (Moon et al., 2020), which assumed conversations occurring between a user and an assistant in a situated, co-observed VR scene. This newer iteration of the dataset lifts the limitations of SIMMC 1.0 by further capturing the complexity of multi-modal conversations: whereas SIMMC 1.0 had at most three objects in a simple, sanitized scene, SIMMC 2.0 provides a far richer visual context with 19.7 objects on average that are often occluded, cluttered, or even out of view. An example dialog is shown in Figure 1.

\(^2\)Dataset is publicly available at https://github.com/facebookresearch/simmc2

The SIMMC 2.0 dataset consists of 11,244 dialogs split into train (65%), dev (5%), devtest (15%), and teststd (15%) sets. Each dialog includes multiple turns where each turn has grounded multi-modal context and an accompanying scene with referential indices. We shall denote a SIMMC dialog with \(r\) rounds as \(D := \{(U_t, A_t, M_t, S_t, B_t)\}_{t=1}^{T}\), where \(U_t\) is user utterance, \(A_t\) system utterance, \(M_t\) multi-modal context, \(S_t\) scene context, and \(B_t\) user belief state at turn \(t\). Here, \(M_t\) is a set of object indices mentioned by the system and \(S_t\) contains the corresponding attributes and locations of all the objects in a scene. User belief state \(B_t\) is composed of dialog act (i.e. user intent) and slot (i.e. a tuple of \((slot\ name, value)\), for instance \(\text{"price", "$11.99"}\)). We also define the dialog history at some turn \(T \leq r\) as \(H_T := \{U_0, A_0, M_0, \ldots, U_{T-1}, A_{T-1}, M_{T-1}\}\).

The assistant needs to make predictions conditioned on history \(H_T\), current user utterance \(U_T\), and the scenes up to the current turn \(S_{t \leq T}\). The object set consists of fashion and furniture domain, where each domain has 288 and 57 items respectively. The system is allowed to look up which item is present in a scene at all time along with its bounding box information. As a side information, the metadata of each object are provided: its non-visual attributes such as brand, size,
rating and price are available for both training and inference, but looking up the visual attribute (e.g. color, pattern, materials, sleeve length) is prohibited for inference so as to make the agent reason with multi-modal information.

3.2 Subtasks

Multi-modal disambiguation (MM-Disamb)
The first subtask is to identify whether the assistant should disambiguate mentions in the next turn given the dialog and multi-modal context. For instance, given user utterance "How much is the pair on the left?", there may be more than two pairs of pants on the left. In this case, ambiguity in reference should be resolved. This can be cast into a binary classification task, and the performance is measured by accuracy.

Multi-modal coreference resolution (MM-Coref)
The second subtask is to map the referential mentions of the user utterance to the object indices in the scene. These mentions should be resolved through the linguistic context and the multi-modal context. The performance is measured by object slot F1 score.

Multi-modal dialog state tracking (MM-DST)
The third subtask extends the traditional uni-modal DST to ground user belief state on the multi-modal objects. This will measure the assistant’s understanding throughout each dialog, which includes disambiguation and coreference resolution. The performance is measured by the F1 score for dialog act and slots.

Response retrieval & generation The last subtask is to retrieve or generate appropriate system utterance. Response generation is evaluated with BLEU-4 (Papineni et al., 2002). For response retrieval, the system is expected to choose the most relevant response from a pool of 100 candidate responses. Recall@k (k ∈ {1, 5, 10}), mean rank, and mean reciprocal rank (MRR) are used for retrieval evaluation.

4 Multi-Modal Transformer Model

The setting of the dataset is similar to that of VQA where finetuning the pretrained VL models are prevalent; however, these models are usually pretrained on natural images (Lin et al., 2014; Krishna et al., 2017) and requires a large number of training samples of 3D rendered images that are aligned properly with text. Hence, we decide to work primarily with pretrained LM and convolutional vision encoder to suit the setting of SIMMC 2.0. In particular, we integrate the visual modality by encoding each object with finetuned ResNet-34 (He et al., 2016). We also index each object in the scene by their referential IDs (canonical object IDs), which are concatenated with corresponding visual representations for subtasks.

In order to further align the different modalities, we provide additional supervision signals at train time by looking up the object metadata. We note that all of the subtasks are related to each other. For example, if the assistant decides that the user utterance needs to be disambiguated, then the appropriate system action is to respond along the line of “Which one are you referring to?”. Once disambiguated, the user may ask for the price of “blue striped shirt”, where representations learned from MM-Coref prediction subtask (and/or attribute classification) can help the model predict the correct slot values for MM-DST and response generation.

We expect that the latent representation of the multi-modal dialog learned from other subtasks will translate readily to other subtasks. Hence, we utilize hard parameter sharing (Caruana, 1993) on the encoder to jointly learn on all subtasks. This reduces not only the number of network parameters, but also the risk of overfitting (Baxter, 1997).

Moreover, we decide to view MM-Coref as a type of set prediction (Zaheer et al., 2017), where joint learning of set cardinality and state distribution has been shown effective (Rezatofighi et al., 2018). Hence, we define an additional empty coreference target prediction (Empty-Coref), a simplified cardinality prediction task that outputs whether the current user utterance has no MM-Coref targets. Moreover, we perform a supervised learning on object attributes to help align object-language modalities.

We adopt BART (Lewis et al., 2019) as the pre-trained language backbone to harness the power of NLU/NLG capabilities demonstrated by pre-trained transformer encoder-decoder. We attach classification heads for MM-Disamb and MM-Coref subtasks to the encoder and LM head for MM-DST and response generation to the decoder. We also perform retrieval by computing the dot product between representation vectors of response candidates and multi-modal dialog context. The overview of the model is provided in Figure 2.
4.1 Input Representation

For all of the subtasks, we define our input to be a simple concatenation \( x := [H_T; U_T; S_{<T}] \) with separators. We define \( H_T \) to be the dialog history up to 2 turns to limit the length of input, i.e., \( \{U_{T-2}, A_{T-2}, M_{T-2}, U_{T-1}, A_{T-1}, M_{T-1}\} \). SIMMCC 2.0 assumes that utterances may mention objects that are not in the current scene \( S_T \) but in the previously observed scene \( S_{<T} \neq S_T \). Hence, our model integrates the objects from the previous scene that are not in the current scene. An exemplar input is provided in Table 1.

4.1.1 Canonical object ID token

A canonical object ID token takes the form of `<\d+>` (e.g., `<32>`). This provides a relational context of the object within the scene, grounding each object to its scene object index provided in the dataset. This scheme was also used in the baseline (Kottur et al., 2021), but without any association to object attributes. In our method, this token intends to provide contextual information of the object alongside its visual attributes, allowing the assistant to make connections between different modalities.

For the assistant to understand the spatial information, we must incorporate location of each object. We follow the commonly used techniques in VL models (Li et al., 2020; Chen et al., 2020; Zhang et al., 2021) for encoding object locations with the bounding box information. Given a bounding box represented by its upper-left and lower-right vertices, \((x_1, y_1)\) and \((x_2, y_2)\), with height \(h\) and width \(w\), we encode its location as a normalized tuple \((x_1/w - 0.5, y_1/h - 0.5, x_2/w - 0.5, y_2/h - 0.5, (x_2 - x_1)(y_2 - y_1)/(h \cdot w))\). This is passed through a location embedding layer (a linear layer followed by layer norm) to be added with the canonical object ID token encoding.

4.1.2 Representation of objects

Each item is represented by its visual and non-visual attributes. The visual attributes are provided by the hidden features of the cropped image encoded by ResNet-34 (He et al., 2016). Once fine-tuned by classifying the objects from the train split scenes to their corresponding visual attributes, the
We formulate MM-Disamb as a binary classification task with randomly initialized learnable embeddings. Both types of attributes are mapped by a linear layer then concatenated to represent an object to match the dimension of BART. For the competition setting (or deployment within virtual environment) where the object identity is readily available at inference, we replace then train the entire object representation except that we use a single-tower architecture. We denote the visual feature of the current object through a binary classification on the pooled output of the encoder from the current scene objects. We also mark those from the previous scene with separator tokens to show how we handle out-of-view objects. The generation target is a concatenation of multiple modalities, which is used by the decoder. The response candidate is prepended.

Table 1: Example input representations for our model. We show only up to last 1 turn due to space limit. Thus, the vision encoder is fixed throughout training of the actual dialog system. The non-visual attributes are randomly initialized learnable embeddings. Both types of attributes are mapped by a linear layer then concatenated to represent an object to match the dimension of BART. For the competition setting (or deployment within virtual environment) where the object identity is readily available at inference, we replace then train the entire object representation with randomly initialized learnable embeddings.

4.1.3 Separator tokens
We define several separator tokens to delimit different components of the multi-modal dialogs. We use \(<\text{SOM}\)\>, \(<\text{EOM}\)\> for the start and the end of multi-modal context and \(<\text{SOO}\)\>, \(<\text{EOO}\)\> for the start and the end of scene objects. Within the scene context, \(<\text{OBJ}\)\> token is used as a marker between current scene objects. We also mark those from the previous scene with \(<\text{PREV_OBJ}\)\>. For generation target, we mark the start and the end of the user belief state with \(<\text{SOB}\)\>, \(<\text{EOB}\)\>.

4.2 Model Specifics

4.2.1 Binary prediction for MM-Disamb and MM-Coref
We formulate MM-Disamb as a binary classification on the pooled output of the encoder from the pooling token \(<\text{DISAMB}\)\>. The binary head for MM-Disamb should predict true if the current user utterance \(U_T\) needs to be disambiguated and false otherwise.

For MM-Coref, we make binary predictions on all objects in \(S_{t≤T}\). We do so by passing the concatenated canonical object (e.g. \(<\text{11}\)\>) and the representation of each object through a binary classification head. The MM-Coref head will predict true if the current user utterance mentions that object and false otherwise. We use a simple cross-entropy loss for both MM-Disamb and MM-Coref, denoted \(L_{\text{mm-disamb}}\) and \(L_{\text{mm-coref}}\).

4.2.2 Auto-regressive LM for MM-DST and response generation
We also approach MM-DST and response generation subtasks with auto-regressive LM following the recent approaches in end-to-end dialog systems. For MM-DST and response generation, we use the standard left-to-right LM loss (Bengio et al., 2003).

\[
L_{\text{LM}} = \sum_{i=1}^{L} -\log P(\omega_i | \omega_1, \ldots, \omega_{i-1}),
\]

where \(\omega_i\) is the \(i\)-th target token and \(L\) the total length of the target.

4.2.3 In-batch negative samples for retrieval
For response retrieval task, we make use of in-batch negative samples for contrastive learning on similarity metrics, following (Karpukhin et al., 2020) except that we use a single-tower architecture. We treat the system responses of the other samples in the batch formatted according to Table 1 as in-batch negatives. We then pool the input and the response candidate representations via bos token to compute their dot product. We use multi-class cross-entropy loss applied to dot-product similarities, i.e.,

\[
L_{\text{retrieval}} = -\log \frac{\exp(x \cdot a^+)}{\sum_{a^- \in B^-(x)} \exp(x \cdot a^-)},
\]

where \(a^+\) is the positive response sample of the input \(x\) and \(B^-(x)\) the set of in-batch negative
responses (assume $x$, $a^+$, and $a^-$ are pooled representations from the encoder). We formulate the task loss $L_{\text{task}}$ as a linear combination of losses from each subtask.

$$L_{\text{task}} = \lambda_{\text{LM}} L_{\text{LM}} + \lambda_{\text{mm-disamb}} L_{\text{mm-disamb}}$$

$$+ \lambda_{\text{mm-coref}} L_{\text{mm-coref}} + \lambda_{\text{retrieval}} L_{\text{retrieval}}$$

(1)

4.3 Auxiliary Tasks

4.3.1 Binary prediction for Empty-Coref

We define an additional Empty-Coref task, in which the assistant predicts whether the current dialog turn has MM-Coref targets. We find this additional signal for coreference resolution, denoted $L_{\text{empty-coref}}$, is advantageous in boosting MM-Coref performance, a type of set prediction task. Moreover, MM-Coref sometimes predicts targets when there is actually none, so we override any MM-Coref predictions if the Empty-Coref prediction is true (i.e. there is no coreference target). For this, we use $<\text{EMPTY_Coref}>$ for pooling. At training time, we use cross-entropy loss for $L_{\text{empty-coref}}$.

4.3.2 Encoding object attributes

We encode object attributes by providing additional supervision signal during training. We do so by simply training to classify each object to its corresponding visual and non-visual attributes such as color, price, and customer ratings. Each object is represented as a concatenation of its canonical object ID and object features as in MM-Coref (refer to Figure 2). Each attribute head predicts a categorical class for each corresponding object, for example, if an object is a grey jacket, the color-attribute head should predict grey and the type-attribute head jacket.

Let $O_{t \leq T}$ be the set of objects in the scene history, $S_{t \leq T}$. We denote attribute multi-class classification loss $L_{\text{att}}$ for all objects in $O_{t \leq T}$.

$$L_{\text{att}} = \sum_{j \in O_{t \leq T}} \sum_{k \leq T} \sum_{c \in C_k} -\mathbb{1}\{c = y_{jk}\} \log P(c),$$

where $K$ is the number of attributes, $C_k$ the set of all classes of the $k$-th attribute, $y_{jk}$ the label of the $k$-th attribute of the $j$-th object, and $\mathbb{1}\{\cdot\}$ is an indicator function.

As a result, the auxiliary loss $L_{\text{aux}}$ is defined as the weighted sum of attribute loss and empty-coreference prediction loss:

$$L_{\text{aux}} = \lambda_{\text{att}} L_{\text{att}} + \lambda_{\text{empty-coref}} L_{\text{empty-coref}}$$

(2)

In summary, we minimize the sum of the task loss $L_{\text{task}}$ (Equation 1) and the auxiliary loss $L_{\text{aux}}$ (Equation 2).

5 Experiments

5.1 Experimental Setup

The details on training hyperparameters are provided in Appendix A. For model selection, we evaluate the model on the devtest split at every 1000 training steps. We give priority to the left-most metric for each task (Table 2) and early stop on those winning the most among 5 subtasks (counting response generation and retrieval separately).

5.2 Baselines

The dataset organizers provided two baseline models: an end-to-end GPT-2 (Radford et al., 2019) and multi-modal transformer networks (MTN) (Le et al., 2019). The baselines handle each subtask separately, except for MM-Coref, MM-DST, and response generation. The GPT-2 baseline generates the user belief state, coreference objects indices, and response in an end-to-end manner given a dialog history with multi-modal context provided in terms of object indices. For retrieval, generated response is compared against available pool of response candidates, from which the candidate with the most likelihood is chosen. MTN baseline conditions on the scene image and dialog history then generate the user belief state and response using a multi-modal transformer. The MTN baseline only implements MM-DST and response generation.

6 Results

The results on the devtest (validation) and teststd (test) splits are shown in Table 2 and 3, respectively. On devtest set, our proposed model outperforms the baselines by a large margin. In the competition setting, we replaced visual feature extractor with object embeddings and scaled the model to BART-large. This model was ranked at the first place with 75.8% coreference F1 in MM-Coref and was declared winner in the response retrieval subtask with 71.2% R@1, 95.0% R@5, 98.2% R@10, and 1.9 mean rank. Despite the simple approach we have taken for representing the multi-modal context, we were able to achieve competitive results with a single model.

For comparison, the winning entry for MM-Disamb and MM-DST, Entry #5, uses separate models, namely RoBERTa-large (Liu et al., 2019)
whose vision encoder is replaced with randomly initialized learnable embedding for identifying objects. To adopt the image encoder to the SIMMC 2.0 architecture with text encoder (RoBERTa-large) \[\text{(Ours)}\] #3 93.9% 84.5% 6.6

Table 2: The official leaderboard of DSTC10 on the teststd set. The subtask winners are bold-faced and runner-ups together to match against the dialog context for MM-Disamb and BART for generating MM-Coref and MM-DST using the same prompt of the baselines without the use of visual features. Even though injecting continuous visual features (as extracted by vision models) may introduce noise for generation, they certainly help with MM-Coref subtasks as some entries achieving more than 65% object F1 utilize visual features \[\#6\] and \[\#7\]. Entry \#8 enumerate visual attributes in the form of natural language tokens without relying on actual visual features.

Table 3: The official leaderboard of DSTC10 on the teststd set. The subtask winners are bold-faced and runner-ups are marked with \[\dagger\], \[\cdot\] means that the entry did not participate in that subtask. Our entry uses 24-layer BART-large whose vision encoder is replaced with randomly initialized learnable embedding for identifying objects. All of the response retrieval entries modify the baseline approach, where the generated response (not the dialog context) is compared against the response candidate pool by different measures. Entry \#1 uses cosine similarity for retrieval score instead of cross-entropy. Entry \#2 uses cross-entropy, but generate from BART. Entry \#4 uses OSCAR \[\text{(Li et al., 2020)}\] with self-supervised few-shot learning for predicting object tags, which act as anchor between image (object) and text (dialog) modalities. The generated response with attached decoder are then compared in the same way as the baseline.

for MM-Disamb and BART for generating MM-Coref and MM-DST using the same prompt of the baselines without the use of visual features. Even though injecting continuous visual features (as extracted by vision models) may introduce noise for generation, they certainly help with MM-Coref subtasks as some entries achieving more than 65% object F1 utilize visual features \[\#6\] and \[\#7\]. Entry \#8 enumerate visual attributes in the form of natural language tokens without relying on actual visual features.

Entry \#6 \[\text{(Lee and Han, 2021)}\] is a multi-tower architecture with text encoder (RoBERTa-large) and image encoder (DeiT) \[\text{(Touvron et al., 2021)}\]. To adopt the image encoder to the SIMMC 2.0 domain, it is contrastively pretrained by matching object image to its natural language attributes and scene (background) image to dialog context. Then, the objects and scene representations are added together to match against the dialog context for MM-Coref prediction. Entry \#7 \[\text{(Huang et al., 2021)}\] encodes object information (index, location, and image) extracted by CLIP \[\text{(Radford et al., 2021)}\] and BUTD \[\text{(Milewski et al., 2020)}\] then inputs the flattened object representations to UNITER \[\text{(Chen et al., 2020)}\] along with dialog context and scene image. MM-Coref predictions are made in terms of binary classification, similar to our approach.

\(\dagger\) \[\text{MTN baseline performs only MM-DST and response generation. The second block provides ablation studies on BART-base, where top two rows are the results of the complete model.} \text{FT}:\ \text{finetuning visual encoder beforehand, JT:}\ \text{subtask joint training,}\ \text{AC:}\ \text{attribute classification loss, EC:}\ \text{Empty-Coref loss, AX:}\ \text{all auxiliary subtasks (attribute classification and Empty-Coref).} \text{For MM-Coref performance of the model without joint training, we report both the results of baseline-like generation (left) and our classification approach (right).
6.1 Ablation Studies

We ablate finetuning of vision encoder, joint training, and auxiliary objectives from BART-base to see how different components of the model affects the performance. Because our model maps the frozen visual features to the model via learnable linear layer, finetuning of the vision encoder before the actual training of the dialog system helped improve the model performance overall. We also observe that joint learning is a crucial part of the success of our approach, as training each subtask separately degrades the performance even more than not finetuning the vision encoder. The effect of different subtask loss coefficients are given in Table 5 of Appendix A.

Removing the auxiliary subtasks affect the MM-Coref performance drastically. First, removing attribute classification loss drops the MM-Coref performance by roughly 13.3%. It also affects the response retrieval performance partially because of the joint-learning approach. Removing Empty-Coref loss degrade the object F1 score slightly, but slightly improves slot F1 as in ablating attribute classification. Removing all of auxiliary subtasks shows even clearer picture, where MM-Coref performance degrades by 20% among other subtasks.

6.2 Visualizing attention

We visualize the learned attention between the two different modalities. Figure 3 shows attention heatmaps from the fifth head in last encoder layer. The rows indicate extracted utterance from $[H_T; U_T]$ and the columns object IDs in in $S_{t<T}$. Table 4 lists the visual-metadata of these objects. According to the visualization, the model was able to make a connection between natural language attributes mentioned in the dialog and the corresponding objects.

<table>
<thead>
<tr>
<th>fashion object ID</th>
<th>color</th>
<th>type</th>
<th>pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>169</td>
<td>light grey</td>
<td>jacket</td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>black, white</td>
<td>blouse</td>
<td>vertical</td>
</tr>
<tr>
<td>256</td>
<td>black</td>
<td>sweater</td>
<td>knit</td>
</tr>
<tr>
<td>168</td>
<td>maroon</td>
<td>dress</td>
<td>plain</td>
</tr>
<tr>
<td>258</td>
<td>brown</td>
<td>dress</td>
<td>plain</td>
</tr>
<tr>
<td>283</td>
<td>purple</td>
<td>dress</td>
<td>plain</td>
</tr>
<tr>
<td>277</td>
<td>grey</td>
<td>trousers</td>
<td>heavy stripes</td>
</tr>
<tr>
<td>115</td>
<td>grey, white</td>
<td>jacket</td>
<td>twin colors</td>
</tr>
<tr>
<td>167</td>
<td>blue</td>
<td>jacket</td>
<td>plain</td>
</tr>
<tr>
<td>005</td>
<td>black</td>
<td>blouse</td>
<td>velvet</td>
</tr>
<tr>
<td>069</td>
<td>black, white</td>
<td>blouse</td>
<td>spots</td>
</tr>
<tr>
<td>265</td>
<td>blue</td>
<td>jeans</td>
<td>denim</td>
</tr>
<tr>
<td>188</td>
<td>blue</td>
<td>trousers</td>
<td>plain</td>
</tr>
</tbody>
</table>

Table 4: Visual metadata of object IDs shown in Figure 3.

7 Conclusion

In this paper, we propose a multi-modal task-oriented dialog system based on BART that can perform all SIMMC 2.0 subtasks at once. Our model integrates the multi-modality of the challenge by utilizing vision features from a vision model. In addition to joint learning of all subtasks, we introduce auxiliary tasks to aid in subtasks. We observe that the joint-learning and other components are crucial in building a successful multi-modal assistant for SIMMC 2.0. Our model is able to perform competitively in all of the subtasks, setting a high bar for the new generation of multi-modal task-oriented dialog systems. Despite the success in SIMMC 2.0, our approach has a few limitations. First, it relies on metadata for non-visual attributes, which may not generalize if a new set of domain items are introduced at inference. Our method also fails to fully capture the semantic locality of objects within the scene (e.g. on the table, in the closet, etc.). We believe that these limitations will be addressed in future works.


A Implementation Details

A.1 Training Hyperparameters

Our model is built on top of BART from HuggingFace (Wolf et al., 2019). We finetune the model for 10 epochs with an initial learning rate of 5e-5 and a batch size of 16 with AdamW optimizer (Loshchilov and Hutter, 2018). We also use linear warmup schedule with 8000 warmup steps and clip gradient norms at 1.0. For decoding, we use top-p sampling (Holtzman et al., 2020) with \( p = 0.9 \) to generate the user belief state and system response.

A.2 Joint Learning Coefficients

We train the model jointly on the sum of Equation 1 and Equation 2. We find the optimal combination of coefficients via grid search with the following choice of coefficient, while fixing \( L_{\text{LM}} \) to 1.0 and grouping MM-Disamb and auxiliary losses together to reduce the search space. Table 5 shows the results of grid search with the final choice of hyperparameters.

- \( \lambda_{\text{mm-disamb}}, \lambda_{\text{att}}, \lambda_{\text{empty-coref}} \in \{0.1, 0.3\} \)
- \( \lambda_{\text{mm-coref}} \in \{0.8, 1.0\} \)
- \( \lambda_{\text{retrieval}} \in \{0.2, 0.4, 0.8\} \)

In general, we see that increasing \( \lambda_{\text{mm-disamb}}, \lambda_{\text{att}}, \lambda_{\text{empty-coref}} \) does not help the model in terms of performance. We also see some performance degradation in MM-Coref as \( \lambda_{\text{retrieval}} \) increases; however, increasing \( \lambda_{\text{mm-coref}} \) improves the overall performance of the model.

A.3 Task-Specific Heads

Object related classification heads (MM-Coref, attribute classification) have input dimension of twice the model dimension of BART (i.e. 2048 for bart-large). For MM-Disamb and Empty-Coref classification head, we use a single linear layer with softmax activation. For MM-Coref and attribute classification, we use an intermediate layer with the same hidden size as the input dimension, which is followed by a linear layer with softmax activation.

B Qualitative analysis

A successful multi-modal agent should be able to recommend objects that fit the user’s requested criteria within the scene context, understand the locations of the objects, and provide the requested information on the object such as ratings and price. We qualitatively analyze the generated system utterances to check whether our model can capture the object attributes along with spatial information.

B.1 Recommending objects from scene

Refer to Table 6 for examples. Upon inspecting generated samples, we observe that our model is often able to recommend appropriate objects that fall under the user’s criteria. The first example take place in a scene with jackets with the color attributes mentioned by the system-generated \( A_T \), demonstrating the ability to capture object attributes. The second example demonstrates the case where the system correctly recommend and ground jacket to the correct location.

However, it is not hard to find cases where the system is able to recommend the correct objects but in a wrong location. The third example demonstrates such case. All of the three recommended objects match those in the ground-truth response, but the system believes that they are all at a different location when in fact they are all on the left wall. We conjecture that our method of encoding object locations did not provide enough spatial information especially because we do not integrate the store structure itself. The retrieved \( A_T \) with the same dialog yield the correct response since all negative samples in the candidate pool did not contain all of the three objects mentioned in the ground truth.

B.2 Predicting coreference object and attributes

Refer to Table 7 for examples. We see that the model successfully identifies which objects and slots are being queried. In most cases, the model outputs the exact corresponding object information without having to lookup the object metadata directly. Furthermore, the model correctly identifies the turn for disambiguation. However, for more complicated instances such as the third example, the model mixes up the reference mentions and identifies the wrong value for the attribute. We also provide examples of all subtasks results (MM-Disamb, MM-Coref, MM-DST, response generation & retrieval) with the corresponding VR scene in Figure 4, 5, 6, 7, and 8.

\footnote{https://github.com/huggingface/ transformers}
Table 5: Hyperparameter grid search on the joint learning coefficients on BART-base. The coefficients are given in the first column in the form of 
$((\lambda_{\text{mm-disamb}}, \lambda_{\text{att}}, \lambda_{\text{empty-coref}}), \lambda_{\text{mm-coref}}, \lambda_{\text{retrieval}})$. Bolded row shows the most optimal hyperparameters in terms of model selection criteria.

<table>
<thead>
<tr>
<th>#1 Disamb</th>
<th>#2 MM-Coref</th>
<th>#3 MM-DST</th>
<th>#4-1 Res. Retrieval</th>
<th>#4-2 Res. Gen</th>
<th>BLEU-4 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Obj. F1 (%)</td>
<td>Slot F1 (%)</td>
<td>Act. F1 (%)</td>
<td>MRR (%)</td>
<td>R@1 (%)</td>
</tr>
<tr>
<td>(0.1, 0.8, 0.2)</td>
<td>91.8%</td>
<td>71.4%</td>
<td>81.5%</td>
<td>94.8%</td>
<td>75.7%</td>
</tr>
<tr>
<td>(0.1, 0.8, 0.4)</td>
<td>91.2%</td>
<td>69.5%</td>
<td>80.0%</td>
<td>94.4%</td>
<td>77.9%</td>
</tr>
<tr>
<td>(0.1, 0.8, 0.8)</td>
<td>92.4%</td>
<td>64.9%</td>
<td>76.2%</td>
<td>92.7%</td>
<td>75.4%</td>
</tr>
<tr>
<td>(0.1, 1.0, 0.2)</td>
<td>92.5%</td>
<td>71.9%</td>
<td>82.0%</td>
<td>95.2%</td>
<td>74.5%</td>
</tr>
<tr>
<td>(0.1, 1.0, 0.4)</td>
<td>92.3%</td>
<td>69.9%</td>
<td>83.2%</td>
<td>93.3%</td>
<td>76.9%</td>
</tr>
<tr>
<td>(0.1, 1.0, 0.8)</td>
<td>91.8%</td>
<td>63.6%</td>
<td>78.7%</td>
<td>93.2%</td>
<td>74.2%</td>
</tr>
<tr>
<td>(0.3, 0.8, 0.2)</td>
<td>92.4%</td>
<td>69.6%</td>
<td>77.9%</td>
<td>95.7%</td>
<td>74.5%</td>
</tr>
<tr>
<td>(0.3, 0.8, 0.4)</td>
<td>92.6%</td>
<td>67.0%</td>
<td>75.4%</td>
<td>95.2%</td>
<td>74.7%</td>
</tr>
<tr>
<td>(0.3, 0.8, 0.8)</td>
<td>91.5%</td>
<td>61.7%</td>
<td>77.1%</td>
<td>93.9%</td>
<td>74.3%</td>
</tr>
<tr>
<td>(0.3, 1.0, 0.2)</td>
<td>91.7%</td>
<td>70.6%</td>
<td>83.2%</td>
<td>95.7%</td>
<td>73.6%</td>
</tr>
<tr>
<td>(0.3, 1.0, 0.4)</td>
<td>92.6%</td>
<td>67.3%</td>
<td>75.1%</td>
<td>95.4%</td>
<td>72.7%</td>
</tr>
<tr>
<td>(0.3, 1.0, 0.8)</td>
<td>92.3%</td>
<td>63.9%</td>
<td>75.3%</td>
<td>94.6%</td>
<td>73.8%</td>
</tr>
</tbody>
</table>

Table 6: Comparison between Generated $A_T$ and ground truth ($GT$) in the case of recommendation. The differences between the generated and the ground-truth (in terms of semantics) are bolded.

$U_T$: Can you show me a jacket that goes with my wardrobe?

Generated $A_T$: There are several jackets which match your criteria, including a black jacket, a grey and white jacket, and a light grey jacket.

$GT$ $A_T$: There are a couple of jackets throughout the store that you might like. What do you think?

$U_T$: Can you recommend a jacket to match my clothes?

Generated $A_T$: Do you like the blue one on the bottom row of the wall? It is a jacket.

$GT$ $A_T$: Do you like the brown one on the top left of the wall or the blue one directly under it? They are jackets.

$U_T$: Can you help me find a cheap blouse?

Generated $A_T$: What do you think of the grey and white blouse on the wall, the maroon, white, and blue on the rack, or the white and grey on the back wall?

$GT$ $A_T$: There are three options on the left wall: a grey and white, a white and grey, and a maroon, white, and blue?

Table 7: Comparison between Generated and $GT$ $A_T$ in the case of disambiguation and informing object attributes. The differences between the generated and the ground-truth (in terms of semantics) are bolded.

$U_T$: What is the rating and price of that gray and white shirt?

Generated $A_T$: Which one?

$GT$ $A_T$: Which one?

$U_T$: The gray and white one hanging on the left.

Generated $A_T$: The grey one is $239.99 and the brown one is only $59.99.

$GT$ $A_T$: Sure! The brown one is $59.99 and the grey one is $239.99.

$U_T$: The green and white blouse in the closet and the red, white, and yellow blouse on the rack.

Generated $A_T$: The former is from Downtown Consignment and is available in S, XL, and XXL. The latter is from Pedals & Gears and is in stock in XS, S, and XL.

$GT$ $A_T$: The first is available in XS, S, and XL and is from The Vegan Baker. The other is in XS, S, and XL and is from Downtown Consignment.
Figure 4: Comparison between the model prediction and the ground-truth of a SIMMC 2.0 dialog instance.
Can you help me find a cheap blouse?

There are three options on the left wall: a grey and white, a white and grey, and a maroon, white, and blue.

"What do you think of the maroon, white, and blue blouse on the left, the grey and white on the right, or the dark grey on the back wall?"

None of the options match that description.

Sorry, but no matches were found. What else can I do for you?

Both have long sleeves.

Of course. Both have been added.

Of course, both have been added to your cart.

Figure 5: Comparison between the model prediction and the ground-truth of a SIMMC 2.0 dialog instance.
Figure 6: Comparison between the model prediction and the ground-truth of a SIMMC 2.0 dialog instance.
<table>
<thead>
<tr>
<th>Utterance</th>
<th>MM-Disamb</th>
<th>MM-Coref</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U0</strong> I’m looking to buy some luxury jeans that have reviewers falling head over heels.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT A0</strong> Those two jeans in the back to the left are a match! Do either of those suit your preferences?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicted</strong> A0: Those two jeans in the back to the left are a match! Do either of those suit your preferences?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>U1</strong> Yeah, I could get on board with those. Which one is more popular with buyers?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT B1</strong> REQUEST:COMPARE, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicted</strong> B1: REQUEST:COMPARE, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT A1</strong> The lighter pair is more highly rated, with a nearly unbeatable average customer rating of 4.9. The darker pair also comes highly recommended, with an average score of 4.6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Retrieved</strong> A1: The lighter pair is more highly rated, with a nearly unbeatable average customer rating of 4.9. The darker pair also comes highly recommended, with an average score of 4.6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>U2</strong> Good to know. Can I also get you to look for something in the same vein as that violet blouse hanging up to the right or the green, violet, and pink one near it, but by the brand Downtown Consignment?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT B2</strong> ASK:GET, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicted</strong> B2: ASK:GET, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT A2</strong> That brand makes the green and white shirt hanging above the light-wash jeans to the left.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Retrieved</strong> A2: That brand makes the green and white shirt hanging above the light-wash jeans to the left.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>U3</strong> I mean that red and white one to the right by the purple shirt and the grey one hanging near the green shirt to the left.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT B3</strong> ASK:GET, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicted</strong> B3: ASK:GET, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT A3</strong> The red and white jacket is rated a 3.3, while the grey jacket is rated a 3.6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Retrieved</strong> A3: The red and white jacket is rated a 3.3, while the grey jacket is rated a 3.6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>U4</strong> Woops, I’m not certain which jackets you’re talking about. Could you please be more specific?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT B4</strong> ASK:GET, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicted</strong> B4: ASK:GET, slots: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GT A4</strong> The red and white jacket has an average customer rating of 3.3, while the grey jacket is rated a 3.6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Retrieved</strong> A4: The red and white jacket has an average customer rating of 3.3, while the grey jacket is rated a 3.6.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Comparison between the model prediction and the ground-truth of a SIMMC 2.0 dialog instance.
Models | Utterance | MM-Disamb | MM-Coref
---|---|---|---
$U_0$ | Are there any highly rated pants around here? | GT | Pred
GT $B_0$ | REQUEST:GET, slots: [type: trousers, customerReview: good] | GT | Pred
Pred $B_0$ | REQUEST:GET, slots: [type: trousers, customerReview: good] | GT | Pred
GT $A_0$ | The first two pairs on the left have great reviews! | GT | Pred
Generated $A_0$ | I think you may like the blue pair on the left and the two grey pairs on the right. | GT | Pred
Retrieved $A_0$ | The first two pairs on the left have great reviews! | GT | Pred
$U_1$ | Okay the last thing I need right now is more tshirts, but I can’t help myself. I’m totally eyeing that white one in the back to the left and the black one hanging on the top row. Which of those is less expensive, and what sizes are those? | GT | Pred
GT $B_1$ | REQUEST:COMPARE, slots: {} | GT | Pred
Pred $B_1$ | REQUEST:COMPARE, slots: {} | GT | Pred
GT $A_1$ | The black shirt pictured here is an extra large, and it sells for $24.99. The white shirt pictured here is a large, and it is priced slightly higher, at $29.99. | GT | Pred
Generated $A_1$ | The white shirt is less expensive, at $54.99, and it is a small. The black shirt costs $29.99 and is an extra large | GT | Pred
Retrieved $A_1$ | The black shirt pictured here is an extra large, and it sells for $24.99. The white shirt pictured here is a large, and it is priced slightly higher, at $29.99. | GT | Pred
$U_2$ | Gotcha. Got anything else for me to see? | GT | Pred
GT $B_2$ | INFORM:REFINE, slots: [type: trousers, customerReview: good] | GT | Pred
Pred $B_2$ | INFORM:REFINE, slots: [type: tshirt] | GT | Pred
GT $A_2$ | My apologies, but there are no more highly rated trousers for you to view at this time. | GT | Pred
Generated $A_2$ | I apologize, but those are the only shirts I can find here. Is there something else I might be able to help you with? | GT | Pred
Retrieved $A_2$ | Unfortunately, those are the only ones I have. Can I show you something different? | GT | Pred
$U_4$ | Oh snap that black blouse back there might be just the thing actually. What’s that cost, and who makes it? | GT | Pred
GT $B_4$ | ASK:GET, slots: {} | GT | Pred
Pred $B_4$ | ASK:GET, slots: {} | GT | Pred
GT $A_4$ | Which blouse are you inquiring about? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
$U_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
GT $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh snap that black blouse back there might be just the thing actually. What’s that cost, and who makes it? | GT | Pred
$U_4$ | Oh snap that black blouse back there might be just the thing actually. What’s that cost, and who makes it? | GT | Pred
GT $B_4$ | ASK:GET, slots: {} | GT | Pred
Pred $B_4$ | ASK:GET, slots: {} | GT | Pred
GT $A_4$ | Which blouse are you inquiring about? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
$U_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
GT $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
$U_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
GT $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
$U_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
GT $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
$U_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
GT $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
$U_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
GT $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred
Generated $A_4$ | I regret to inform you there are no similar blouses with that pattern. | GT | Pred
Retrieved $A_4$ | Oh no, it looks like there are no similar blouses in that pattern. What else can I do for you today? | GT | Pred

Figure 8: Comparison between the model prediction and the ground-truth of a SIMMC 2.0 dialog instance.