Multimodal and Joint Learning Generation Models for SIMMC 2.0

Thanh-Tung Nguyen, Wei Shi, Ridong Jiang, Jung-jae Kim

Institute for Infocomm Research, A-STAR
{nguyentt2;shi_wei;rjiang;jjkim}@i2r.a-star.edu.sg

Abstract

There is more and more interest in research of building an AI assistant that can communicate with human in a multimodal conversational setting. Most existing task-oriented dialog datasets do not coordinate the dialog in the user's multimodal context. The Situated Interactive MultiModal Conversations datasets (SIMMC 1.0 and 2.0) were introduced recently to allow research to train machines to consider not only the dialog history but also the scene content. The datasets provide fully annotated dialogues where the user and the agent see the same scene elements and the agent can be decisive to update the scene. This year's SIMMC 2.0: Situated Interactive Multimodal Conversational AI challenge, held as the third track in the Tenth Dialog System Technology Challenge, pushes the development of many methods on SIMMC 2.0 dataset. In this paper, we present our approaches in five subtasks: disambiguation classification, multimodal coreference resolution prediction, dialog state tracking, response generation and response retrieval. For subtasks 1, 3 and 4, we combine all the outputs into a single string and adapt BART-based encoder-decoder framework to make the predictions. For subtask 2, we propose Bi-encoder & Poly-encoder model to match the visual objects with the dialogue turns. For subtask 5, we applied BART-based framework to find more relevant responses among given candidates. Our models came in second place for the response retrieval subtask and a good place for the others. Our models' performance in both the public dev-test and the private test-std datasets shows the robustness of our approaches.

Introduction

In recent years, building an AI assistant, which can process the conversational context the same way as a human assistant, has received more attention from AI research communities. To match the human performance, the AI assistant needs to understand the multimodal context of the conversation beyond the traditional natural language processing settings (Crook et al. 2019; Moon et al. 2020; Gunasekara et al. 2020). This Natural Language Understanding (NLU) process requires taking into consideration of past conversation and external information (meta data or the scene both user and the assistant observe). Moreover, the assistant needs to respond to the user in a reasonable manner in terms of syntax and semantics. Following this direction, the SIMMC challenges in the Dialog State Technology Challenge (DSTC) provide datasets to develop a multimodal shopping assistant. The scenario in this year's SIMMC 2.0 consists of the dialogue between a user and a shopping AI assistant with the close-to-real-world context. In particular, each dialogue is associated with a virtual reality scene of a store, which includes the objects that can be mentioned in the conversation. This is different from the SIMMC dataset last year, where the virtual reality environment does not reflect the real user situations at a store. The dataset also provides object information (meta data) to allow researchers to model context better.

The SIMMC 2.0 challenge consists of 5 subtasks: 1) disambiguation classification, 2) multimodal coreference resolution prediction, 3) dialog state tracking, 4) response generation and 5) response retrieval. We model the subtasks 1, 3 and 4 as a joint learning task. Instead of dealing with each task separately, we represent the output of each task as a string and concatenate all of them into one single output string. This is inspired by recent trend of adopting pretrained models for multitask learning (Moon et al. 2020; Raffel et al. 2020; Kottur et al. 2021; Huang et al. 2021). In particular, we utilize the encoder-decoder framework, where the encoder captures the past multimodal conversational interaction between a user and a virtual assistant while the decoder generates the combined outputs of the three subtasks. One crucial advantage of the joint learning is that it allows us to train on multiple subtasks while sharing a common encoder-decoder. Intuitively, diambiguation, state tracking and response generation can benefit from each other: A plausible state tracking may help generate a plausible response. This multi-task setting cannot be possible in non-neural methods where classification and generation tasks are considered separately. For the response retrieval task, we use the encoder of the encoder-decoder framework. As the response candidates do not have disambiguation label or dialogue state tracking, we train only on the response data. The model encodes the conversational context, outputs the conditional probabilities of the candidates and uses the probabilities to rank the candidates. This model and the multitask model we described before are trained and evaluated independently.

Given one or more scenes shared between the virtual assistant and a user, Sub-Task #2 aims to predict the ID(s) of the objects mentioned in the current user transcript. For example, in the transcript "Do you have anything with a similar size as the brown shirt and the grey and brown?", the mentioned objects IDs are "[40, 12]", which correspond to the IDs of "the brown shirt" and "the grey and brown" in the current scene. A scene is a generated VR scene screenshot. A scene contains multi-modal contexts, with an image showing all the objects that appear in the snapshot and a JSON file describing the name, ID, bounding box and position of the objects and their relationships. In addition, there is another JSON file containing the metadata (e.g., price, available sizes, color, pattern) of all the objects. We regard the task as a ranking task, and build a Bi-encoder (Humeau et al. 2020) and Poly-encoder (Humeau et al. 2020) model to get the similarity score between each candidate object and the current user transcript. And we select the top-K ranked objects as the predicted objects. Bi-encoder encodes the input context and a candidate using two transformers, which are initially started with the same weights in the pre-training, but are updated separately during fine-tuning.

We make our code available at https://github.com/i2r-simmc/i2r-simmc-2021/

Task Description

In this section, we describe more details of the SIMMC 2.0 datasets and challenge. The datasets contain annotated multimodal dialogues generated using finite-state-machines as user and system simulators in a multi-modal framework, and then paraphrased by human annotators.. Each dialogue corresponds to a shopping experience, in which the user communicates with the assistant in natural language while co-observe a scene of the shop. This is different from the existing work in multimodal dialogs (Das et al. 2017; Hori et al. 2018; de Vries et al. 2018; Gunasekara et al. 2020) and Visual Question Answering (Anderson et al. 2018) where there are primary and secondary observations. The user can observe the scene's objects in multiple viewpoints to make a request but does not know detailed information of the objects. On the other hand, the assistant has access to meta information of objects, which include unique properties like size, brand, cost and position of the objects within the scene. Therefore, a virtual assistant not only needs to consider the dialogue history but also keeps track of the environment state to comprehend the user's request and produce the relevant response.

The SIMMC 2.0 datasets have a total of 11K dialogs and about 117K utterances with 1.5K unique commercial store scenes in fashion/furniture domain. The fashion dataset is established from 3D digital assets of real-life clothing objects (e.g. pants, trousers) while the furniture dataset is based on the 3D digital furniture assets from an e-commerce site. These assets/objects are combined with seed scenes from respective domains to get the pool of scenes. The scenes are finally manipulated by using random camera view. A dialogue starts with a user request and asks for recommendation from the assistant. Each dialogue turn has human and system transcripts annotated dialogue acts/intents, references of mentioned objects to the bounding box in the respective scene. This rich multimodal context allows a systematic study of the diverse expression in multimodal dialog.

Following the previous challenge, SIMMC 2.0 datasets are also split into four subsets: train, dev, dev-test and std-test. The train and dev sets, as the names suggest, are used for training and validation processes. The dev-test set is used to evaluate the model performance publicly and outside of the challenge while the test-std set is not available publicly and is used for official evaluation of the challenge. Overall, the metrics for evaluation processes are computed in a task specific manner for both dev-test and std-test. We report our models' performance on both sets.

We briefly go through all sub-tasks in the challenge:

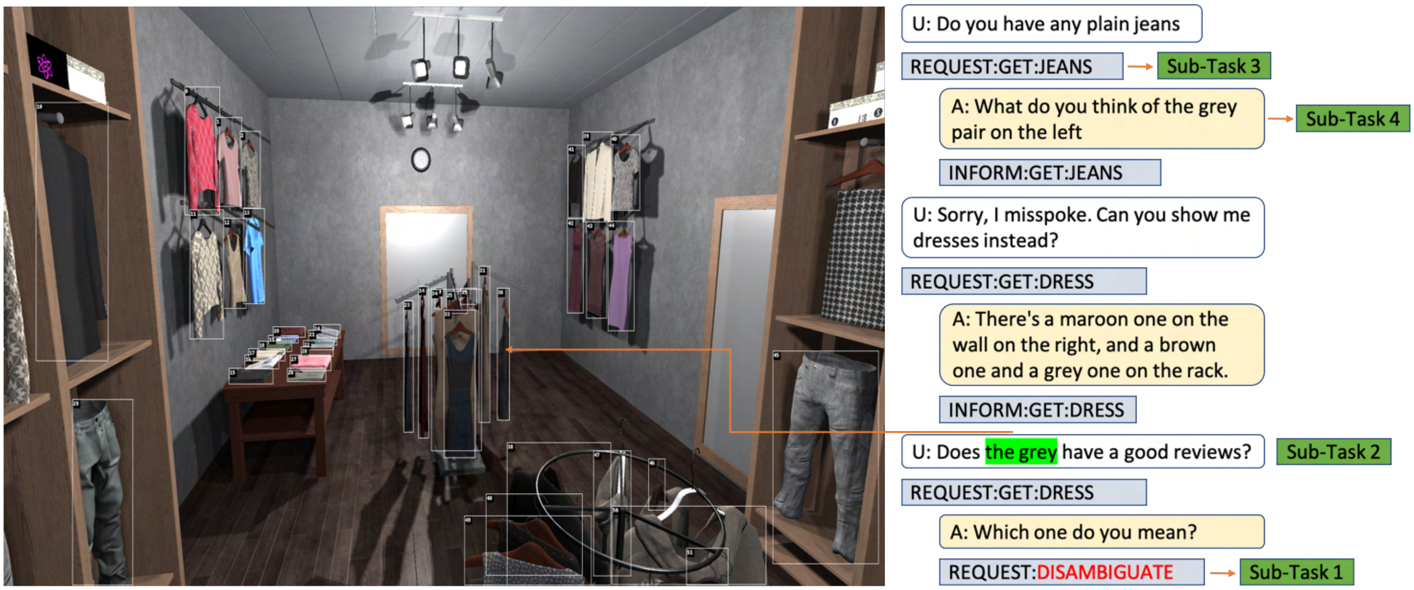
- Sub-task 1 - Multimodal Disambiguation: The first sub-task's goal is to classify whether the user utterance contains any ambiguous reference that the assistant agent should disambiguate. This is a binary classification task whose input are current user utterances, previous assistant turns, multimodal context (scene and objects). The evaluation metric is binary classification accuracy. Moreover, the metric is computed only for some specific turns instead of entire dialogue.

- Sub-task 2 - Multimodal Coreference Resolution: The second sub-task is to resolve the objects referred to in a user utterance and return their canonical IDs that were defined in each scene. In addition to the similar input like sub-task 1, this coreference resolution task also makes use of the ground-truth bounding boxes that segment objects in each scene. Moreover, the previous scenes are also considered to make the prediction as users may mention/refer to an object that they are mentioned from viewpoints/scenes in the past. Coreference Precision, Recall and F1 are evaluation metrics for this sub-task.

- Sub-task 3 - Multimodal Dialog State Tracking: The third sub-task is to track the dialogue acts and the associated slot pairs for multimodal conversation. This sub-task measures the assistant's understanding of users request through each turn. This sub-task also uses the similar input as sub-task 2. The evaluation metrics for this sub-task consists of intent accuracy and slot precision, recall and F1.

- Sub-task 4 - Response Generation: The fourth sub-task aims to generate assistant responses to the user requests. Based on user utterances, ground-truth belief states and multimodal objects/scenes, the assistant needs to reply in a reasonable manner. For the generation task, the traditional metric BLEU-4 score is used to measure the performance. Moreover, the metric is computed only for the last turn in each dialog.

- Sub-task 4 - Response Retrieval: The final sub-task is to retrieve assistant response to the user request from a list of 100 candidates which are generated randomly and uniquely. The performance is evaluated based on the recall@1 (R@1), recall@5 (R@5), recall@10 (R@10), mean rank, and mean reciprocal rank (MRR).



U: Do you have any plain jeans

REQUEST:GET:JEANS

A: What do you think of the grey pair on the left

INFORM:GET:JEANS

U: Sorry, I misspoke. Can you show me dresses instead?

REQUEST:GET:DRESS

A: There's a maroon one on the wall on the right, and a brown one and a grey one on the rack.

INFORM:GET:DRESS

U: Does the grey have a good reviews?

REQUEST:GET:DRESS

A: Which one do you mean?

REQUEST:DISAMBIGUATE

Figure 1: Illustration of all sub-tasks in SIMMC 2.0 dataset. The right-hand side is an example of a dialog between a user and an assistant. The left hand side is the corresponding shopping scene. For each of user's requests, the assistant detects any disambiguation (sub-task 1), matches the ambiguous reference in the text to the object in the scene (sub-task 2), tracks the dialog state (sub-task 3) and generates or retrieves an appropriate response (sub-task 4).

Methods

Baseline Models

The challenge organizers benchmark the datasets by adopting two baseline models:

- Multimodal Dialog State Tracking (MM-DST) by (Moon et al. 2020) that fine-tunes GPT-2 (Radford et al. 2019) to generate the belief states and assistant response (sub-tasks 3 and 4). Specifically, this baseline takes previous dialog context and flattened object/scene information as the input, which is similar to causal language model approach (Peng et al. 2020; Hosseini-Asl et al. 2020). While being effective for utilizing a large pre-trained language model, the drawback in this model is that the flattened string may lose the structure information in its original format data (e.g. {pattern: striped, color: blue}) and give wrong signal to the model.
- Multimodal Transformer Network (MTN) (Le et al. 2019) that encodes features from scenes/object snapshots and dialog together. While being effective to combine image and language features, this model is not pretrained and needs to train from scratch with the challenge datasets only. This may not be comparable to adopting the large pre-trained language models to the datasets.

Joint Learning Model

In this part, we describe our approach to solve sub-tasks 1,3,4 altogether. Let $\mathcal{D}$ denote a dialog in SIMMC dataset with n turns. Formally, $\mathcal{D}$ can be expressed as:

$$\mathcal{D} := \{((U_i, A_i, M_i, B_i)\}_{i=1}^{n} \qquad (1)$$

where U_i, A_i are the user and assistant utterances at turn i, M_i is the multimodal context that includes scene, object representations in form of text (i.e., object metadata) or image and B_i is the belief state which consists of disambiguation label (B_i^{DL}; sub-task 1), object references (sub-task 2), and intent and slot (B_i^{IS}; sub-task 3).

Inspired by recent work to apply natural language generation technique on non-generation tasks (Raffel et al. 2020; Brown et al. 2020), we combine the outputs of Disambiguation Classification, Dialog State Tracking, Response Generation (sub-tasks 1, 3 and 4) into one single output. Then the model input and output at turn i are

$$\text{input}_i = U_{\leq i} \text{ SEP}_1 A_{<i} \qquad (2)$$

$$\text{output}_i = B_i^{DL} \text{ SEP}_2 \ B_i^{IS} \text{ SEP}_3 \ A_i \qquad (3)$$

where $U_{\leq i}$ indicates current and previous user utterances, $A_{<i}$ previous assistant utterances, and SEP$_1$, SEP$_2$, SEP$_3$ are special tokens for separation between different inputs and outputs. One thing to note here is that belief state and assistant response appear in the entire dialog while disambiguation labels only appear in a portion of them. Therefore, we allow the dummy case that a dialog turn does not have disambiguation label and let the respective B_i^{DL} is empty.

As we can combine the three subtasks into one single generation task, we also adopt the encoder-decoder framework to this problem. The probability of the output $\boldsymbol{y}$ for a given input $\boldsymbol{x}$ can be expressed as:

$$\begin{aligned} P_\theta(\boldsymbol{y}|\boldsymbol{x}) &= P_\theta(B^{DL}, B^{IS}, A|\boldsymbol{x}) \\ &= P_\theta(B^{DL}|\boldsymbol{x})P_\theta(B^{IS}|\boldsymbol{x}, B^{DL})P_\theta(A|\boldsymbol{x}, B^{DL}, B^{IS}) \end{aligned}$$

This shows advantage of our joint learning model: It generates outputs for the three separate tasks at the same time

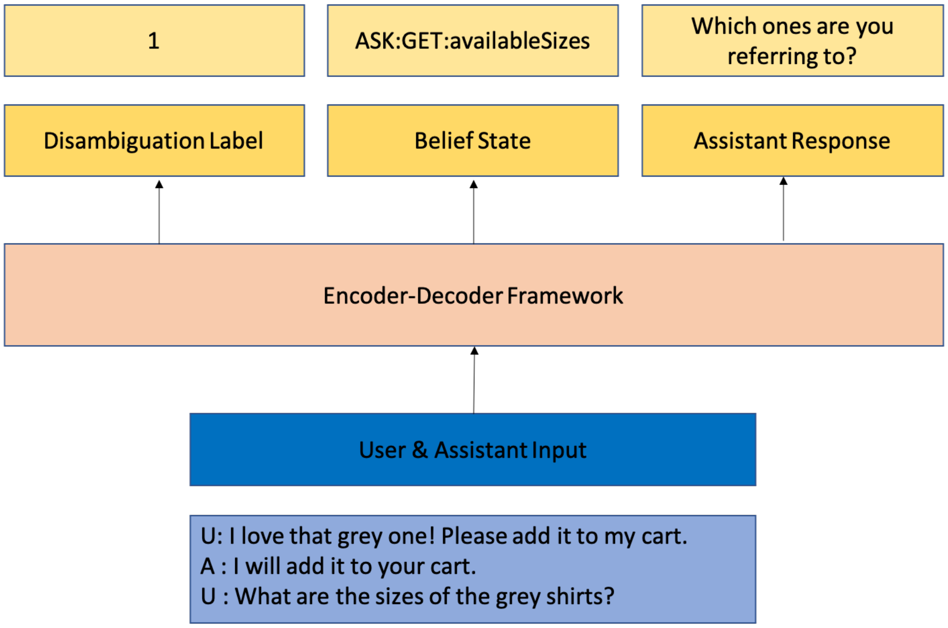

1
ASK:GET:availableSizes
Which ones are you referring to?
Disambiguation Label
Belief State
Assistant Response
Encoder-Decoder Framework
User & Assistant Input
U: I love that grey one! Please add it to my cart.
A : I will add it to your cart.
U : What are the sizes of the grey shirts?

Figure 2: Illustration of generating the three sub-tasks' outputs in a single string.

while sharing a common encoder. Intuitively, all the tasks can benefit from each other as the assistant can generate good response given it already knows whether the user utterance has any disambiguation or any intent/slot act. Following the conditional probability formula, our model accumulates the knowledge from the simplest task (sub-task 1) to the most complex task (response generation).

Model Architecture

We now describe the architectural components of our model: the encoder and decoder. Overall, we apply the Transformer based model (BART) (Lewis et al. 2020) to SIMMC datasets. While fine-tuning the model, we only train to minimize the cross entropy loss and exclude the reconstruction loss in the original model.

Dialog Encoder Given an input sequence of source text $\boldsymbol{x} = (x_1, \dots, x_n)$ that combines previous user and assistant utterances as equation (2), we use Transformer Encoder (Vaswani et al. 2017) to encode it into a continuous space representation. The input is first passed through the embedding layers and then through multi-attentive blocks. In the encoding process, each multi-attentive block is computed using self attention mechanism. We take the last layer hidden states to represent the input text:

$$\begin{aligned} \boldsymbol{e}_x &= \text{Embedding}(\boldsymbol{x}) \\ \boldsymbol{h}_x &= \text{Self-Attention}(\boldsymbol{e}_x) \end{aligned} \tag{4}$$

Dialog Decoder Our model uses an uni-directional Transformer Decoder as the decoder. We generate the output (equation 3) from left to right. At each decoding step t, the decoder applies cross attention on the source text $\boldsymbol{h}_x$ and self attention on the previous states $\boldsymbol{d}_{<t}$ to generate the current state $\boldsymbol{d}_t$. Then $\boldsymbol{d}_t$ is fed into a soft-max layer to get the probabilities for the generated token t

$$\boldsymbol{d}'_t = \text{Cross Attention(Self-Attention } (\boldsymbol{d}_t), \boldsymbol{h}_x) \tag{5}$$

$$\boldsymbol{d}_t = \text{Softmax } (\boldsymbol{d}'_t) \tag{6}$$

Training Objective We train our model by minimizing the total loss defined as:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{generation}}(\theta) + \lambda||\theta||_2^2 \tag{7}$$

where $\mathcal{L}_{\text{generation}}(\theta)$ is the cross entropy loss for generation. we also apply an L_2-regularization on the parameters with λ, the regularization strength.

Response Retrieval Model

Similar to joint learning model, we apply BART encoder-decoder model for sub-task 4. We train our model on generation task with past dialog as input and assistant response as output. The training objective is also to minimize the combination of cross entropy and regularization losses. We use the negative cross entropy of each candidate as the score to rank all candidates in the response pool and choose the candidate that maximizes score:

$$\text{Score}(\boldsymbol{y}_i) = \mathcal{L}_{\text{generation}}(\boldsymbol{y}_i|\boldsymbol{x}) \tag{8}$$

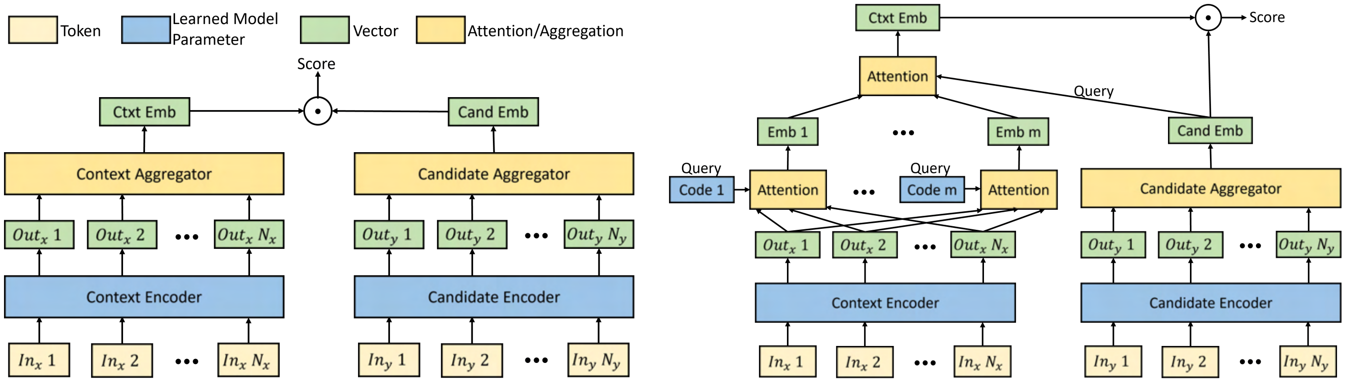
$$i^* = \arg\max \text{Score}(\boldsymbol{y}_i)) \tag{9}$$

Adapt Bi-Encoder & Poly-Encoder for Multimodal Coreference Resolution

Problem Formulation Following the definition as illustrated by equation 1, we elaborate the multimodal context M_i in turn i as

$$M_i = (S_i, O_i, I_i) \tag{10}$$

where S_i is a set of scene text descriptions, including the name, ID, bounding box and position of the objects appearing in the scene and their relationships, O_i is a set of object



(a) Bi-encoder

(b) Poly-encoder

Figure 3: Bi-encoder & Poly-encoder Illustration. (a) Bi-encoder encodes context and candidate separately. (b) Poly-encoder also encodes context and candidate separately, but adds an attention between the global features of the input context and a given candidate to catch the interactions. Adapt from (Humeau et al. 2020)

Data Type	Attributes in Metadata
Furniture	name, brand, customerRating, price, materials
Fashion	name, brand, customerReview, price, size

Table 1: Non-Visual Metadata of objects in Furniture and Fashion data.

metadata, including name, brand, color and other attributes, and I_i is a set of scene images. Table 1 shows the metadata of Furniture and Fashion, respectively. Since visual metadata can't be used, we only list the non-visual metadata.

Sub-Task #2 aims to predict object IDs mentioned in U_i using $\sum_{i=1}^{i} U_i$, $\sum_{i=1}^{i-1} A_i$ and M_i for turn i, $i = \{1, ..., n\}$.

Bi-encoder As Figure 3a shows, in Bi-encoder, the input context and a candidate are encoded by two transformers, separately. The output vectors of context and candidate are:

$$y_{ctxt} = red(T_1(ctxt)) \qquad y_{cand_i} = red(T_2(cand_i)) \tag{11}$$

where $ctxt$ is the input context, $cand_i$ is a candidate, T_1 and T_2 are two transformers, $T(x) = h_1, ..., h_N$ is the output of a transformer T, and $red(\cdot)$ is a function that reduces sequence of vectors into one vector.

Poly-encoder As Figure 3b shows, Poly-encoder also encodes input context and candidate using two separate transformers, which is the same as Bi-encoder. However, in order to catch the interactions between context and candidates, Poly-encoder makes two updates. It indicates that the input context is usually much longer than a candidate, so Poly-encoder uses m global feature vectors $y^1_{ctxt}, ..., y^m_{ctxt}$ to represent the input context, while Bi-encoder only uses one vector. Specifically, y^i_{ctxt} is represented as:

$$y^i_{ctxt} = \sum_j w_j^{c_i} h_j \tag{12}$$

where $(w_1^{c_i}, ..., w_N^{c_i}) = softmax(c_i \cdot h_1, ..., c_i \cdot h_N)$, c_i is context code, $i = 1, ..., m$. The m context codes are initialized randomly and learnt during fine-tuning. Poly-encoder then attends m global features on each candidate vector and gets the representation of the input context as:

$$y_{ctxt} = \sum_i w_i y^i_{ctxt} \tag{13}$$

where $(w_1, ..., w_m) = softmax(y_{cand_i} \cdot y^1_{ctxt}, ..., y_{cand_i} \cdot y^m_{ctxt})$.

Training Objective of Bi-encoder & Poly-encoder We compute the dot-product $s(ctxt, cand_i) = y_{ctxt} \cdot y_{cand_i}$ as the score of a candidate $cand_i$ given the input context $ctxt$. We train our Bi-encoder and Poly-encoder model by minimizing the cross entropy loss:

$$\begin{aligned}\mathcal{L}_{\text{rank}}(cand_i|ctxt) = &-\log s(ctxt, cand_1) \\ &+ \sum_{i=2}^{M} \log s(ctxt, cand_i)\end{aligned}$$

where $cand_1$ is the ground-truth candidate object and $cand_i$, $i = 2, ..., M$ is the negative candidate objects.

Adapt Bi-encoder & Poly-encoder A dialog contains several turns. For each turn, there are one or more corresponding scenes. For turn i, we regard the scenes appearing in turn $1, ..., i$ as the current scenes since it may mention previous scenes. Since Sub-Task #2 aims to predict the object ID(s), we need to obtain candidate objects from the current scenes. We regard all the objects appearing in the current scenes as the candidate objects. In the original setting of Bi-encoder and Poly-encoder, each input context has only one ground-truth candidate label. But in Sub-Task #2, there are usually more than one mentioned object for each user transcript. So in the training process, if a user transcript contains N mentioned objects, we will generate N samples for the transcript. And each sample only contains one mentioned object in the candidate objects, all the other

Datasets	Model	Sub-task 1	Sub-task 3		Sub-task 4
		Accuracy	Slot F1	Intent F1	BLEU
dev-test	MM-DST	73.9	81.72	94.53	0.192
	MTN (Le et al. 2019)	-	74.75	93.40	0.22
	Our model	88.8	90.0	96.28	0.34
test-std	MM-DST	73.5	83.8	94.1	0.20
	MTN (Le et al. 2019)	-	76.7	92.8	0.21
	Our model	89.5	87.8	96.2	0.26

Table 2: Results of joint learning models in SIMMC 2.0 dev-test & test-std sets.

Datasets	Model	Sub-task 4				
		R@1	R@5	R@10	Mean Rank	MRR
dev-test	Random	0.010	0.050	0.100	50.0	0.052
	GPT2	0.026	0.107	0.184	38.0	0.088
	MM-DST	0.404	0.64	0.73	11.91	0.516
	Our model	0.558	0.799	0.876	5.6	0.666
test-std	MM-DST	0.428	0.654	0.749	11.0	0.535
	BART + supervision signal	0.712	0.95	0.982	1.9	0.815
	Our model	0.496	0.747	0.845	6.6	0.612

Table 3: Results of response retrieval models in SIMMC 2.0 dev-test & test-std sets.

mentioned objects will be filtered in the candidate objects, in order to guarantee that each training sample only contains one ground-truth object.

We add the keyword "System :" before system transcript, "User :" before user transcript and "Objects :" before the mentioned object IDs in the previous system transcript annotation. We represent input context in turn i as $ctxt_i$ = "$System\ :\ A_{i-1}\ SEP\ Objects\ :\ [list\ of\ object\ IDs]\ SEP\ Users\ :\ U_i$".

We represent a candidate object Obj_K (K is the object ID) as $cand_{Obj_K}$ = "$K\ \ O_i(Obj_K)\ \ S_i(Obj_K)$". $O_i(Obj_K)$ is the non-visual metadata of object Obj_K, containing a set of attributes showed in Table 1. We add attribute name before attribute value. For example, if the "brand" of a candidate object is "Modern Arts", we represent the attribute as "brand : Modern Arts". For $S_i(Obj_K)$, we use "prefab_path", "unique_id", "index", "bbox", and "position" of object Obj_K.

Following Bi-encoder and Poly-encoder, we compute the dot product between y_{ctxt} and each y_{cand_i}, and ranking the candidate objects based on the score in the descending order. Finally, in the test process, we select top-K ($K = 2$ in our experiments) candidate objects as the predicted objects.

Experiments

Joint Learning Model

Setup We follow the train/dev/dev-test/test-std splits provided by the challenge organizers. This results in 7,307 dialogs for training, 563 for development, 1,687 for dev-test and 1,685 for test-std. For our model setup, we use BART-large encoder-decoder models with 24-layer, 16-head Transformer Encoder and 24-layer, 16-head Transformer Decoder. The model's hidden size is 1,024. The beam width is 4. Our model is trained using Adam optimizer (Kingma and Ba 2015) with a batch size of 24 turns. Our learning rate is initialized at 0.00001. We also follow a learning rate schedule that increases learning rate linearly in 200 warm-up steps. Model selection for testing is based on the cross entropy loss on the development set. We train the models on one single Tesla V100 GPU.

Results The experimental results on SIMMC 2.0 dev-test and test-std sets are shown in Table 2. As it can be seen, our model achieves an accuracy of 88.8% and 89.5% in the dev-test and test-std splits of sub-task 1, outperforms the baseline by 24.9% and 16% respectively. In sub-task 3, our model also achieves slot F1 scores of 90% and 87.8%, exceeding the baselines by 9.3% and 4%. The result in Intent F1 also outperforms the baselines by 1.8% and 2.1%, reaching 96.3% and 96.2% respectively. In sub-task 4, our models outperform the baselines by 11 BLEU in dev-test set and 5 BLEU in test-std set.

Response Retrieval Model

Setup We used the same model settings as adopted in our joint learning models.

Results The experimental results on SIMMC 2.0 dev-test and test-std sets are summarized in Table 3. In dev-test set, we achieve the average recall@1 of 55.8 %, mean rank 5.6 and MRR 0.666 which outperform the baselines. In test-std set, we have similar results which show our model is consistent across different evaluation sets. We achieve the second place in the challenge in this task.

Ablation Study

To validate our multimodal decisions, we performed an ablation study in SIMMC 2.0 sub-tasks. In particular, we evaluate (*i*) the impact of joint learning vs single training models, and (*ii*) whether we should combine the text dialog context with the text representation of the scene (non-visual meta data). We follow the same setup as Joint Learning Model and compare only the results in dev-test set (because test-std is not public).

Joint Learning vs. Single model Table 4 shows results for single models and joint learning model. We can see that the joint learning model outperforms all the other models in respective sub-tasks. This gives the joint learning model significant advantages as it performs not only better than the single sub-task models but also requires less parameters.

Model	**Sub-task 1**	**Sub-task 3**		**Sub-task 4**
	Accuracy	Slot F1	Intent F1	BLEU
Sub-task 1 only	87.1	-	-	-
Sub-task 3 only	-	87.2	91.2	-
Sub-task 4 only	-	-	-	0.30
Our model	88.8	90.0	96.3	0.34

Table 4: Results of joint learning model and single models in SIMMC 2.0 dev-test set.

Model	**Sub-task 1**	**Sub-task 3**		**Sub-task 4**
	Accuracy	Slot F1	Intent F1	BLEU
Meta-data	87.7	24.9	95.40	0.31
Our model	88.8	90.0	96.28	0.34

Table 5: Results of our model with text-only inputs & with all the inputs including meta-data in SIMMC 2.0 dev-test set.

Normal Input vs. Meta Data Table 5 shows results for the models with text-only inputs (Normal Input) and all the inputs including metadata (Meta-data). For the latter, we concatenate the text inputs with the text representation of the scene into a single input string. We can see that it improves upon the single tasks but is still worse than our joint learning model with text-only inputs. It shows that the we need a better way to incorporate the metadata.

Bi-Encoder and Poly-Encoder Model for Multimodal Coreference Resolution

Setup We use the BERT-base (Devlin et al. 2019) model to implement Bi-encoder and Poly-encoder[1]. Specifically, we use uncased BERT of which the number of transformer layers is 4, the hidden size is 512 and the number of attention head is 8. We set the maximum length of the input context as 128 and the maximum length of a candidate object as 256. And our model is trained using Adam optimizer with batch size of 32, epoch number of 10 and learning rate of 0.00005. We train the models on one single Tesla V100 GPU.

Results Table 6 shows the results of our Bi-encoder and Poly-encoder models on Sub-Task #2 using SIMMC 2.0 dev-test and test-std datasets. Bi-encoder model performs slightly better than Poly-encoder model on dev-test. We thus submitted the Bi-encoder model for the evaluation with test-std, and so the performance of the Poly-encoder model on test-std is not reported.

Model	**Object F1**	
	dev-test	test-std
GPT2	0.366	0.441
Bi-encoder	0.405	0.422
Poly-encoder	0.392	–

Table 6: Results of Bi-encoder and Poly-encoder on SIMMC 2.0 dev-test and test-std.

Conclusion

In this paper, we have presented our approaches for the third track of DSTC 10 Challenge (SIMMC 2.0) We proposed a total of three models: a joint learning BART-based model that encodes the dialog context to generate the combination string of disambiguation classification, dialog state tracking and response generation; Bi-encoder and Poly-encoder models that encode and match the input context and object candidates; BART-based model to rank the probabilities of response candidates given dialog context to retrieve the assistant response. Through experiments, we have shown that our methods outperform the baseline models and our models achieved the second place for response retrieval sub-task on test-std. In the future, we would like to extend our models so that they can capture multimodal information more efficiently.

Acknowledgement

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3674–3683.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell,

[1] https://github.com/chijames/Poly-Encoder

A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Crook, P. A.; Poddar, S.; De, A.; Shafi, S.; Whitney, D.; Geramifard, A.; and Subba, R. 2019. SIMMC: Situated Interactive Multi-Modal Conversational Data Collection And Evaluation Platform. *arXiv preprint arXiv:1911.02690*.

Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D. 2017. Visual dialog. In *CVPR*.

de Vries, H.; Shuster, K.; Batra, D.; Parikh, D.; Weston, J.; and Kiela, D. 2018. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 4171–4186. Association for Computational Linguistics.

Gunasekara, C.; Kim, S.; D'Haro, L. F.; Rastogi, A.; Chen, Y.-N.; Eric, M.; Hedayatnia, B.; Gopalakrishnan, K.; Liu, Y.; Huang, C.-W.; et al. 2020. Overview of the ninth dialog system technology challenge: Dstc9. *arXiv preprint arXiv:2011.06486*.

Hori, C.; Cherian, A.; Marks, T. K.; and Metze, F. 2018. Audio Visual Scene-Aware Dialog Track in DSTC8. *DSTC Track Proposal*.

Hosseini-Asl, E.; McCann, B.; Wu, C.-S.; Yavuz, S.; and Socher, R. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Huang, X.; Tan, C. S.; Ng, Y. B.; Shi, W.; Yeo, K. H.; Jiang, R.; and Kim, J. J. 2021. Joint Generation and Bi-Encoder for Situated Interactive MultiModal Conversations. *AAAI 2021 DSTC9 Workshop*.

Humeau, S.; Shuster, K.; Lachaux, M.; and Weston, J. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *8th International Conference on Learning Representations, ICLR*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kottur, S.; Moon, S.; Geramifard, A.; and Damavandi, B. 2021. SIMMC 2.0: A Task-oriented Dialog Dataset for Immersive Multimodal Conversations. *arXiv preprint arXiv:2104.08667*.

Le, H.; Sahoo, D.; Chen, N.; and Hoi, S. 2019. Multimodal Transformer Networks for End-to-End Video-Grounded Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5612–5623.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Moon, S.; Kottur, S.; Crook, P. A.; De, A.; Poddar, S.; Levin, T.; Whitney, D.; Difranco, D.; Beirami, A.; Cho, E.; Subba, R.; and Geramifard, A. 2020. Situated and Interactive Multimodal Conversations. *arXiv preprint arXiv:2006.01460*.

Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Gao, J. 2020. SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model. *arXiv preprint arXiv:2005.05298*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 5998–6008. Curran Associates, Inc.