Multimodal Context Carryover

Prashan Wanigasekara, Nalin Gupta, Fan Yang, Emre Barut, Zeynab Raeesy,
Kechen Qin, Stephen Rawls, Xinyue Liu, Chengwei Su, Spurthi Sandiri

Alexa AI-Natural Understanding, Amazon

{wprasha, nalgupta, fyaamz, ebarut, raeesyzr,
qinkeche, sterawls, luxnyu, chengwes, spurthi}@amazon.com

Abstract

Multi-modality support has become an integral part of creating a seamless user experience with modern voice assistants with smart displays. Users refer to images, video thumbnails, or the accompanying text descriptions on the screen through voice communication with AI powered devices. This raises the need to either augment existing commercial voice only dialogue systems with state-of-the-art multimodal components, or to introduce entirely new architectures; where the latter can lead to costly system revamps. To support the emerging visual navigation and visual product selection use cases, we propose to augment commercially deployed voice-only dialogue systems with additional multi-modal components. In this work, we present a novel yet pragmatic approach to expand an existing dialogue-based context carryover system (Chen et al., 2019a) in a voice assistant with state-of-the-art multimodal components to facilitate quick delivery of visual modality support with minimum changes. We demonstrate a *35%* accuracy improvement over the existing system on an in-house multi-modal visual navigation data set.

1 Introduction

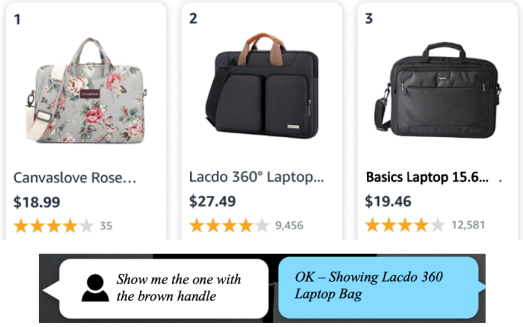

Figure 1: Product Selection Use Case

Tracking the state of the conversation and understanding context is a crucial component in voice-based dialogue systems. The Context Carryover

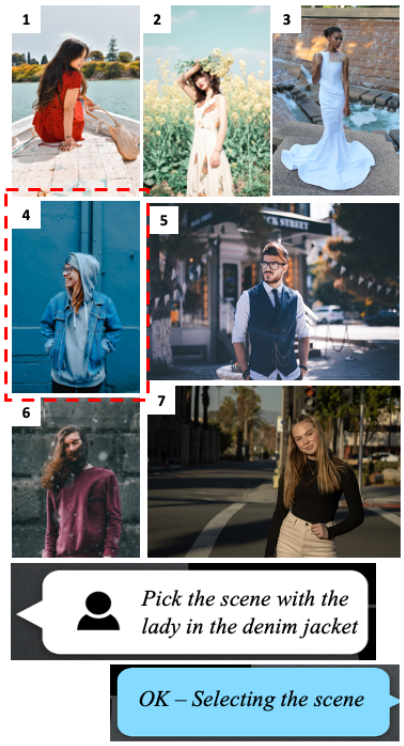

Figure 2: Scene Selection Use Case. The images are from unsplash.com and used here only for illustrative purposes.

Figure 3: Video Selection Use Case

(CC) framework (Chen et al., 2019a; Naik et al., 2018; Sharaf et al., 2018; Rastogi et al., 2019) is a framework that handles identification and carryover of relevant context information in a multi-turn dialog interaction between a voice assistant and the user. The CC framework determines which tokens and intents in the most recent system-user interaction history are relevant as supporting information to fulfill user's current request. The details

of the context carryover framework are well documented in (Chen et al., 2019a). One key limitation of the framework is that it is purely text-based, and therefore it struggles to capture user interactions that involve visual components. In this work, we introduce augmentations that enable the Context Carryover framework to deal with multimodal use cases. We focus on two specific use cases that's related to a user's visual navigation and selection experience: Visual product selection and visual scene and video selection.

Visual product selection, demonstrated in Fig. 1, consists of the use case where the user is referring to a single product on the screen. In the provided example, the user is shopping for a handbag and the voice assistant is displaying a number of handbags on the screen. The user selects one out of many handbags on the screen using a referring utterance, for instance the color of the handbag. The user is free to use any other natural language phrase that can differentiate the product from the others displayed on the screen.

In **visual scene and video selection**, as seen in Fig. 2, and Fig. 3, the user can refer to a scene or movie that has multiple products with a more cluttered visual landscape. Here, a "scene image" is defined as an image of an individual in a landscape wearing multiple products (dress, hat, purse, sunglasses etc.). The user then tries to select a scene using a referring expression (e.g., "The scene with the lady in the denim jacket"). In Fig. 3, a movie can be associated with multiple frames and the user can refer to the movie by referring to an action or content from a specific frame.

The main contributions of our work are:

1. We introduce a **Vision Augmentation** scheme that enables ingestion of visual content in a dialogue-based context carryover framework.

2. We introduce an **Aligned Vision and Text Augmentation** that incorporates the latest state-of-the-art developments in multi-modal contrastive learning to a dialogue-based context carryover framework.

3. The newly proposed methods result in significant accuracy improvements on an in-house data collected through Amazon Mechanical Turk (MTurk). We present sensitivity analyses that display the effectiveness of the various suggested model augmentations on our in-house dataset.

4. We introduce a synthetic data generation pipeline that generates synthetic visual product selection data that helps to train the models and cuts down on manual annotation and MTurk survey costs.

2 Related work

Thanks to the advent of Transformer-based models over the past few years, multimodal representations have seen significant advances. The types of multimodal models can be roughly categorized into three, **a) Single Encoder** (Girshick et al., 2013; Long et al., 2014; Simonyan and Zisserman, 2014; Tan and Bansal, 2019; Chen et al., 2019b; Zhang et al., 2021; Li et al., 2020; Wanigasekara et al., 2022), **b) Dual Encoder** (Radford et al., 2021; Li et al., 2021a; Zhang et al., 2020; Jia et al., 2021; Yuan et al., 2021), and **c) Encoder-Decoder** models (Vinyals et al., 2014; Wang et al., 2021, 2022; Piergiovanni et al., 2022; Li et al., 2022). Attempts at unifying these foundational models have also been made in (Yu et al., 2022; Singh et al., 2021). **Single Encoder** models appear early in the multimodal literature and pave the way for the other 2 types of models. For this family of models, usually the image and text representations exist in separate spaces and there is an ensuing fusion layer. **Dual Encoder** models leverage image-text contrastive loss (Oord et al., 2018; He et al., 2019; Chen et al., 2020; Tian et al., 2019) during training, exhibit higher image-to-text alignment and bring the image, text representations to a common more aligned representation space. They perform well on image-text retrieval tasks but underperform in vision-language understanding tasks requiring higher reasoning, e.g., Visual Question Answering (VQA), and Natural Language Inference (NLI).

Our current task of Multimodal Context Carryover uses the latest advances in the multimodal representation learning space and injects state-of-the-art components with minimal changes into a framework for dialog tracking and slot selection, and results in a system that can handle multimodal user-system dialog interaction. Our current multimodal use cases are set up to be similar to a text-to-image retrieval task that occur within the context of a user-system dialog interaction. Thus, we incorporate the latest developments in **Dual Encoder** design in our work, since the approach is well suited for the multimodal text-to-image re-

trieval step. The latest state-of-the-art models that incorporate multimodal representations to dialogue state tracking systems, e.g., VDST (Pang and Wang, 2019), Flamingo (Alayrac et al., 2022), and VDTN (Le et al., 2022), would require costly system revamps.

Recently, Kottur et al. (2021) released a novel multimodal conversation dataset with labeled dialogue state (e.g., entity and dialogue act), which motivated further studies (Garcia et al., 2022; Agarwal et al., 2021). These datasets contain dialog act and products but do not have the scene and video information required for our purposes. For our current study we resort to collecting our own dataset through Amazon Mechanical Turk which is catered for our commercial needs.

3 Models

3.1 Problem Formulation

Each interaction between the user and the system can be formulated as a sequence of utterances H, consisting of alternating utterances between the user U and the system S: $H = (h_0^U, h_1^S, h_2^U, ..., h_{dist}^{\{U,S\}})$, where each element $h \in H$ is an utterance either by the user, h^U or the system, h^S. We refer to H as the dialog history. A subscript $dist$ denotes the utterance distance, which measures the offset from the most recent user utterance (h_0^U). The i^{th} token of an utterance with distance $dist$ is denoted as $h_{dist}[i]$.

Each utterance in the dialog history, H consists of slots. A slot $x = (dist, k_x, v_x)$ in a dialog is defined as a key-value pair that contains information about an entity. For e.g., in Fig. 1 the user says, "*Show me the one with the brown handle*". Here one of the slots would be [COLOR:Brown]. Each slot is defined by the distance of it's corresponding utterance $dist$, slot key k_x and slot value v_x. We refer to X as the context slots which comprise of all the slots in the dialog history.

In addition to the context slots X which are derived from the dialogs, we also have on-screen lists which can be present in the current turn as shown in Fig. 1. Users can reference items in these lists either through visual features or through references to the title, e.g., "Canvaslove Rose one ...". A list object $l = (k_l, v_l, I_l)$ in the current turn is defined as a key-value pair along with the associated image. The key k_l in our case is *ProductTitle*, the value v_l is the title itself and I_l refers to the image associated with the list object. We refer to L as all the list objects in the current turn.

Given the dialog history H, context slots X and the on-screen list L, we can define the candidate slots as $C = (X \cup L)$. The task can be formulated as correctly identifying the subset of candidate slots C which are relevant to the current turn. A binary decision is made jointly over each of these candidate slots C by the model F_{joint}, which takes the slot interdependencies into consideration, i.e.,: $F_{joint}(C, H) = C_{carry}$, where $C_{carry} \subseteq C$.

The full details of the context carryover (CC) architecture which forms our baseline are provided in Appendix A.1. The baseline solution is not capable of ingesting visual content and hence cannot perform selection based on visual features. In Section 3.2, we introduce the vision and vision aligned text augmentations to the CC model, which add the capability to process visual features and are the main contributions of this paper.

3.2 Augmentations

3.2.1 Vision Augmentation

Recently CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), ALBEF (Li et al., 2021b), ConVIRT (Zhang et al., 2020) train dense, aligned image-text embeddings using contrastive loss. The training requires having N matched $(image, text)$ pairs where the text can be free form. The bidirectional contrastive losses for the i^{th} image-text pair is given in equation 1 and equation 2 in Appendix A.2. The image and text are projected onto a shared embedding space $\mathbf{I} \in \mathbb{R}^d$, $\mathbf{T} \in \mathbb{R}^d$ respectively. $\langle \mathbf{I}_i, \mathbf{T}_i \rangle$ represents the cosine similarity and $\tau \in \mathbb{R}^+$ is a temperature parameter. The losses are then added as seen in equation 3 in Appendix A.2.

Our on-screen image selection use case is slightly different from this generic paired $(image, text)$ training setting. In our case, the user makes a reference to a specific product, scene, or movie that is shown on the screen, which is more akin to a text-to-image retrieval task. The user also focuses on differentiating the desired product from the list of products shown on the screen. This is slightly nuanced than a generic text description of a product as the referring utterance is conditioned on the desired image and other surrounding images.

In our initial solution, we obtain the CLIP vision embeddings for the product images and add it to the CC framework as shown in Fig. 4a) which we term as the **Vision Augmentation**. In Fig. 4a), b), the term **List SeMI** stands for a List of [**Se**]mantically

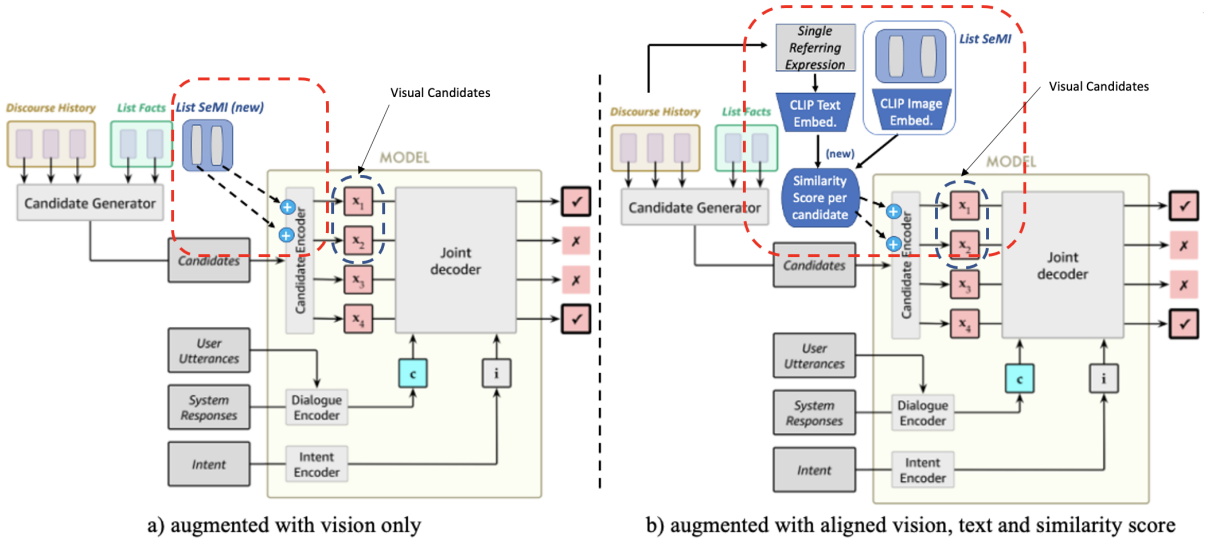

Figure 4: Augmented Context Carryover Models. Vision only augmentation (a), Vision, text and similarity score augmentation (b)

[**M**]eaningful [**I**]mages. More concretely, each product shown to the customer is represented by a list object $l = (k_l, v_l, I_l)$ and considered as a potential carryover candidate as mentioned in Section 3.1. As part of the Vision Augmentation, we use the CLIP visual embedding of the product image as I_l. These product list objects with CLIP visual embeddings are then sent to the CC Candidate Encoder. The CC Encoder-Decoder framework subsequently decides whether to carry over the list product object to the next dialogue state, which would signify a product selection. The rationale here is to simply augment the existing system with the visual modality and evaluate the effectiveness on a multimodal dataset. Even though we select CLIP as our initial vision embedding, the system is compatible with any embedding trained with a contrastive loss (e.g. we also show similar performance improvements on ALBEF (Li et al., 2021b)).

3.2.2 Aligned Vision and Text Augmentation

Using the notation from Section 3.1, at a given moment, there are n product list item objects $l = (k_l, v_l, I_l)$ shown to the user. Here, k_l is the key word *ProductTitle*, the value v_l is the actual title itself and I_l is the associated image. Given the product images $\{I_1, I_2, ...I_n\}$, their titles $\{v_1, v_2, ..., v_n\}$, and the most recent user referring utterance h_0^U which is obtained from the dialog history, our task here is to find the best product list object l that matches the user's request h_0^U. We utilize CLIP to bring images $\{I_1, I_2, ...I_n\}$ and the textual referring expression h_0^U to the same embedding space, and get the dot product similarity between each image in $\{I_1, I_2, ...I_n\}$ and h_0^U, as shown in Fig. 4b). In other words, we obtain the similarity score per candidate list image with the referring utterance. We term this as the **Aligned Vision and Text Augmentation**. The pseudocode for this operation is shown in Fig. 7, in Appendix A.3. The resulting multimodal dot product tensors are shown in Fig. 8, in Appendix A.4.

4 Experiments

4.1 Datasets

In this section, we describe the newly gathered Amazon Mechanical Turk (MTurk) multimodal dataset and the pre-existing text-only dataset.

4.1.1 Multimodal dataset

The MTurk dataset is created by showing Mechanical Turkers (MTurkers) product images, scene images, video thumbnails and asking them to pick one out of the many products, scenes or movies using a single referring utterance. We define one such referring act that includes multiple images and a single referring utterance as a single "instance". The newly collected MTurk dataset has a Train/Dev/Test split sizes of 33,526/4,087/4,152 instances respectively. Additional details about the dataset are included in the Appendix A.5.

We also utilize an internally annotated dataset, that we refer to as the existing Context Carryover dataset, details of which can be seen in Appendix

Table 1: Results on the multimodal test set for various CLIP augmentation schemes and the CC baseline. The + <modality type> indicates an augmentation.

#	Visual Emb.	Model	Slot Level					
			Weighted (P, R, F1)				Accuracy	Δ
			P	R	F1	Δ		
1	None	Baseline	0.59	0.63	0.60	-	0.6301	-
2	CLIP	+ Utterance Text	0.65	0.67	0.54	-10.05%	0.6671	5.87%
3		+ Similarity	0.81	0.80	0.78	30.41%	0.7955	26.25%
4		+ Visual	0.78	0.78	0.78	31.12%	0.7811	23.96%
5		+Visual + Utterance Text	0.79	0.78	0.79	32.26%	0.7844	24.49%
6		{+ Visual + Utterance Text + Similarity } (with non-contrastive fine-tuning)	0.80	0.80	0.79	33.35%	0.8003	27.01%
7		**{ + Visual + Utterance Text + Similarity }**	**0.85**	**0.85**	**0.85**	**42.15%**	**0.8484**	**34.65%**

A.6.

4.1.2 Synthetic Visual Data Generation Pipeline

Figure 5: Synthetic Data Generation

Even though we use MTurk data for our current study, it is expensive to generate and infeasible to extend to more domains. An alternative cheaper and scalable approach to quickly collect carryover data in multimodal settings is to employ a data synthesizer. The synthetic data generation process is as follows; for each synthetic data sample, we first randomly sample a slot type (e.g., bag) and visual attributes (e.g., red) to create slot candidates. Note that for each slot, we only randomly select one type from a pre-defined *object* list, and we sample three different visual attributes under the same attribute category (e.g., color) from a pre-define *attribute* list. For example, we select one slot type bag from the object list, and then we draw three visual attributes red, blue, orange from the attribute list. We then combine them to obtain three slots: red bag, blue bag, orange bag to simulate the screen shown to the user when shopping for bags. In the second step, we retrieve an image from a product image catalog for each generated slot. To do so, we employ CLIP to get text embeddings of each slot: $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$. For each image in the product catalog, we precompute the image embedding with the same CLIP model: $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$. We use the inner product as the similarity metric to perform the image retrieval: $I_x = \arg\max_i \boldsymbol{e}_x^\top \boldsymbol{p}_i$, where I_x denotes the image selected for slot x. To add more randomness, we retrieve the top-k similar images, and randomly select one image from the k candidates. After getting all of the product images (and associated metadata), we simulate a user selection phrase by randomly selecting one out of the three generated slots as the ground-truth and fill it in a predefined template.

4.2 Results

We compare various combinations of vision, text and similarity augmentation schemes against the baseline CC model in Table 1. All the models are trained on a combined CC and multimodal MTurk training dataset. Since we are interested in how our models perform on multimodal use cases, we show the results only on the multimodal test dataset. Further details of the experiment setup, such as model training details, are described in the Appendix A.7.

In Table 1, row 1 is the current baseline CC model, and rows 2-7 are the augmentations. Augmenting the existing CC framework with only CLIP Visual components (Table 1, row 4) gives a 23.96% accuracy improvement over the baseline on the multimodal test set. When adding CLIP Vision and CLIP user current utterance Text embeddings (Table 1, row 5), we see accuracy gains increase to 24.49%. The highest improvement comes when the CLIP vision, text embeddings and the dot product similarity scores (Table 1, row 7) are given to the CC framework with a 34.65% accuracy improvement over the baseline. These methods keep the

Table 2: Performance improvements on MTurk test data when MTurk training data is added to the CC train data set.

#	Model	Train Data	Slot Level					
			Weighted (P, R, F1)				Accuracy	Δ
			P	R	F1	Δ		
1	baseline	CC	0.53	0.67	0.53	-	0.67	-
2	baseline	MTurk	0.68	0.68	0.68	27.81%	0.68	2.34%
3	baseline	CC + MTurk	0.59	0.63	0.60	11.48%	0.63	-5.31%
4	**Visual**	**CC + MTurk**	**0.78**	**0.78**	**0.78**	**46.17%**	**0.78**	**17.39%**

CLIP embeddings frozen, but in Table 1, row 6 we attempt to fine-tune the CLIP embeddings in an end-to-end fashion using the CC framework. We find that fine-tuning the CLIP embeddings in our setting does not provide further gains. This maybe because the generic loss of the CC framework is non-contrastive (i.e., it's cross-entropy based) and thus it does not improve the effectiveness of CLIP embeddings being further fine-tuned. A similar observation is recorded in parallel work Flamingo (Alayrac et al., 2022) and likened to *"catastrophic forgetting"*. In Table 1, row 3, we exclude the vision and textual embeddings altogether and provide only the dot product similarity scores. We find that only providing similarity scores (row 3) is on par with row 5 which is to provide both visual and textual embeddings. This has implications where the CC encoder can simply work with similarity scores instead of embeddings. Finally, in row 2 we only give the CC framework the CLIP Text embeddings (i.e., no vision components) and we find the performance to be worse than the other augmentations. We hypothesize that this is because the CC framework gets textual information from two sources in row 2. One from the dialogue history, which it processes in the usual fashion described in Section A.1 through the CC slot carryover framework, and one via the CLIP Text embeddings. Since there is no accompanying visual input, the CLIP Text input is redundant and might add additional noise, which leads to minor accuracy improvements and a weighted F1 degradation seen in row 2.

We are also interested in seeing how the CC baseline model behaves when the multimodal MTurk training data is added to its training set. More simply put, we want to check whether adding the MTurk training data to the CC framework will improve the performance of the CC baseline on the multimodal MTurk test set, *even if* the CC baseline is a purely text-based system that have no notion of the visual modality. From Table 2, row 1 we see that even when no multimodal training data is added, the CC baseline still has an absolute accuracy of 67% on the MTurk test set. This can be attributed to the CC framework using the product image titles which are textual to make inferences. In Table 2, row 2 when the baseline is only trained on the multimodal MTurk dataset (without the 1.28M pre-existing CC dataset) there is an 2.34% accuracy improvement relative to the baseline, mainly due to training and testing distributions being similar. In Table 2, row 3, when the data sets are combined during training, the baseline shows a −5.31% degradation compared to Table 2, row 1 which indicates that the CC baseline is not equipped to handle a combined multimodal and non-multimodal dataset. In Table 2, row 4 we see that the best results are obtained when visual modality related model changes are added and the model is trained on the combined multimodal and pre-existing CC dataset.

We also experiment with ALBEF (Li et al., 2021b) embeddings and show results in Appendix A.8. We anticipate the science community to produce ever-improving dense multimodal embeddings as time goes on, and hope that our simple yet effective augmentation enables commercial frameworks to utilize the latest state-of-the-art embeddings with minimal changes.

5 Conclusion

We augment the existing Context Carryover framework with Visual and Vision Aligned Text components. We collect a multimodal dataset which mimics real world customer interactions to train and evaluate our models. We show a 35% accuracy improvement when the existing CC framework is augmented with Vision and Vision Aligned Text components.

Limitations

Our solution is only limited to the English language: our training data only contain products with English titles and all of the referring expressions are in English. Transferring the model to another language will require re-training the model and potential architecture changes. Furthermore, the Mturkers who provided the experssions in our study may not be representative of the user demographics, and the data may not provide a well grounded proxy of user behavior. While collecting more data can mitigate some of these limitations, curation and validation of visual expressions is a time-consuming and expensive process, which is why our dataset is limited in size.

Our models leverage visual embeddings from systems such as CLIP, ALBEF which have their own set of limitations. For instance, CLIP is known to fail in cases which requires counting objects, or relations of multiple objects in an image. Thus, visual models leveraging CLIP embeddings will have issues with referring expressions that refer to counts of objects. Further, we need to be cognizant and optimize for inference latency, which prevents us from using large scale language or vision models which could potentially improve upon the current solution.

Ethics Statement

Although our solution has no unethical applications or risky broader impacts, we need to consider aspects of fairness. In our setting, the images shown to the users can contain images of people along with the products. We need to consider how sensitive queries, e.g., ones that refer to protected attributes of the people in the image or expressions that contain hateful or derogatory speech, should be resolved.

During the data collection and model training process we take strong consideration on the type of referring expression we are curating and using to train the model. Expressions that contain references to protected and/or physical attributes of people are filtered out to ensure that our model is not capable of handling sensitive queries.

6 Acknowledgements

The authors would link to thank Balaji Kamakoti and Henry Zhang for their contributions in clearly defining the use cases and driving the visual product selection project. Rohit Parimi for managing the engineering effort. Melanie Gens, Matt Johnson, and Adam Kalman for their contributions in engineering design. Chevanthie Dissanayake for her help in visualizations.

References

Sanchit Agarwal, Jan Jezabek, Arijit Biswas, Emre Barut, Shuyang Gao, and Tagyoung Chung. 2021. Building goal-oriented dialogue systems with situated visual context.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Tongfei Chen, Chetan Naik, Hua He, Pushpendre Rastogi, and Lambert Mathias. 2019a. Improving long distance slot carryover in spoken dialogue systems.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019b. Uniter: Universal image-text representation learning.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale.

Jack FitzGerald, Shankar Ananthakrishnan, Konstantine Arkoudas, Davide Bernardi, Abhishek Bhagia, Claudio Delli Bovi, Jin Cao, Rakesh Chada, Amit Chauhan, Luoxin Chen, Anurag Dwarakanath, Satyam Dwivedi, Turan Gojayev, Karthik Gopalakrishnan, Thomas Gueudre, Dilek Hakkani-Tur, Wael Hamza, Jonathan J. Hüser, Kevin Martin Jose, Haidar Khan, Beiye Liu, Jianhua Lu, Alessandro Manzotti, Pradeep Natarajan, Karolina Owczarzak, Gokmen Oz, Enrico Palumbo, Charith Peris, Chandana Satya Prakash, Stephen Rawls, Andy Rosenbaum, Anjali Shenoy, Saleh Soltan, Mukund Harakere Sridhar, Lizhen Tan, Fabian Triefenbach, Pan Wei, Haiyang Yu, Shuai Zheng, Gokhan Tur, and Prem Natarajan. 2022. Alexa teacher model. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM.

Francisco Javier Chiyah Garcia, Alessandro Suglia, José Lopes, Arash Eshghi, and Helen F. Hastie. 2022. Exploring multi-modal representations for ambiguity detection & coreference resolution in the SIMMC 2.0 challenge. CoRR, abs/2202.12645.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning.

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

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 4903–4912. Association for Computational Linguistics.

Hung Le, Nancy F. Chen, and Steven C. H. Hoi. 2022. Multimodal dialogue state tracking.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. 2021a. Align before fuse: Vision and language representation learning with momentum distillation.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. 2021b. Align before fuse: Vision and language representation learning with momentum distillation.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2014. Fully convolutional networks for semantic segmentation.

Chetan Naik, Arpit Gupta, Hancheng Ge, Mathias Lambert, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. In Interspeech 2018. ISCA.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding.

Wei Pang and Xiaojie Wang. 2019. Visual dialogue state tracking for question generation.

AJ Piergiovanni, Wei Li, Weicheng Kuo, Mohammad Saffar, Fred Bertsch, and Anelia Angelova. 2022. Answer-me: Multi-task open-vocabulary visual question answering.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Mathias Lambert. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers), pages 97–105, Minneapolis, Minnesota. Association for Computational Linguistics.

Amr Sharaf, Arpit Gupta, Hancheng Ge, Chetan Naik, and Lambert Mathias. 2018. Cross-lingual approaches to reference resolution in dialogue systems.

Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2021. Flava: A foundational language and vision alignment model.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive multiview coding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision.

Prashan Wanigasekara, Kechen Qin, Emre Barut, Fan Yang, Weitong Ruan, and Chengwei Su. 2022. Semantic vl-bert: Visual grounding via attribute learning. In 2022 International Joint Conference on Neural Networks (IJCNN), pages 1–8.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. 2021. Florence: A new foundation model for computer vision.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. 2020. Contrastive learning of medical visual representations from paired images and text.

A Appendix

A.1 Existing Context Carryover framework

Our baseline architecture follows a similar approach to Chen et al. (2019a), where they jointly model the slots to make a slot carryover decision. One of the key differences in our model is that we act on entities extracted from an on-screen list along with the entities in the discourse history. We highlight the components of the existing CC model as follows:

Candidate Generation We create candidates based on a handcrafted slot map, which defines carryover compatibility between each pair of slots. We create a set of slots X from the context by leveraging the slot map to identify slots compatible with the current turn slots. We also append the candidates with the on-screen list entities.

Slot Encoder Given a candidate slot, which is represented as a (slotKey, slotValue), we average the word embeddings of the slot pair tokens and convert them into a fixed-length vector representation $\mathbf{x} \in R^{d_x}$.

Dialog Encoder We serialize the tokens in the dialog and use an LSTM (Hochreiter and Schmidhuber, 1997) to create a fixed length embedding. $\mathbf{c}_{dialog} = LSTM(H)$, where $\mathbf{c}_{dialog}$ is the dialog encoding and H is the dialog.

Intent Encoder The intent tagged by an upstream Natural Language Understanding module is encoded by averaging the word embeddings of the tokens to create a fixed length embedding $\mathbf{int} \in R^{d_{int}}$.

Decoder Given the encoded representation of the slots $\{x_1, ..., x_n\}$, dialog $\mathbf{c}_{dialog}$, and intent **int**, we use the self-attention decoder presented in (Chen et al., 2019a). Self-attention allows the decoder to model relationships between all the slots in the dialog, which is shown to yield better results. In our model, we use 12 attention heads, which allows the model to jointly attend to information from different perspectives at different positions.

A.2 Contrastive Learning equations

$$\mathcal{L}_i^{(image \rightarrow text)} = -\log \frac{\exp\left(\langle \mathbf{I}_i, \mathbf{T}_i \rangle / \tau\right)}{\sum_{k=1}^{N} \exp\left(\langle \mathbf{I}_i, \mathbf{T}_k \rangle / \tau\right)}, \tag{1}$$

$$\mathcal{L}_i^{(text \rightarrow image)} = -\log \frac{\exp\left(\langle \mathbf{T}_i, \mathbf{I}_i \rangle / \tau\right)}{\sum_{k=1}^{N} \exp\left(\langle \mathbf{T}_i, \mathbf{I}_k \rangle / \tau\right)}. \tag{2}$$

The final loss is a weighted combination of the two losses averaged over the training dataset. Here $\lambda \in [0, 1]$ is a scalar weight.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \Big(\lambda \mathcal{L}_i^{(image \to text)} + (1 - \lambda) \mathcal{L}_i^{(text \to image)} \Big). \quad (3)$$

A.3 Pseudocode for the vision aligned text dot product

The pseudocode for the vision aligned text dot product is shown in Fig. 7.

A.4 Multimodal dot product tensors

Fig. 8 a) shows the dot product between the utterance text and the product visual embeddings for visual product selection for each carryover candidate. Fig. 8 b) shows the dot product between the utterance text and scene or video and associated product's visual embeddings for visual scene and video selection. Fig. 8 c) shows the dot product between the utterance text and the product metadata text associated with each candidate, where the product metadata can be available for both visual product selection and visual scene and video selection.

A.5 Additional Details on the multimodal dataset.

Some randomly sampled instances from the multimodal MTurk dataset for visual product selection are given in Fig. 6. To better simulate a real-world customer interaction with the voice assistant, the MTurkers are free to use any phrase to refer to the product image. Some MTurkers use specific product attributes like color, size, shape, product material, product label text while there are instances where more ambiguous terms are used (e.g., *"the animal one"*). To dissect the dataset further, we encode a few randomly selected product images and their associated labels in a joint CLIP embedding space.

As seen in Fig. 9a. The product images are shown on the horizontal axis and the product labels are shown on the vertical axis. The numbers in the table are CLIP similarity scores $\in [0, 1]$ between the images and the product labels (higher scores mean higher similarity). Fig. 9a has a dual purpose: first, it shows the products and their labels in a matrix format where products that match with multiple labels or labels that match with multiple products can be clearly seen; second, it shows the effectiveness of CLIP embeddings in terms of quantifying image-text similarity. Ideally, the diagonal elements of the matrix should contain the largest scores, but we can see that there are a few off diagonal high similarity scores, which indicates that there is high ambiguity. In Fig. 9b we look at the alignment between the images, their labels and the referring utterance. It can be clearly seen from Fig. 9b row 1 that the image that matches the referring utterance (*"the black one"*) has the highest CLIP similarity score (the middle image has the highest alignment score; 0.24 with the referring utterance compared to the other two images in row 1).

A.6 Existing Context Carryover Dataset

The existing CC dataset is created by internal annotators who were shown the dialogue history, current turn, context slots and were asked to select all the appropriate slots for the current turn. The dialogs originate from a commercial voice assistant, and we process the data so that users are not identifiable ("de-identified"). The dataset spans 30 domains, 500 intents and includes both within domain (dialog that span a single domain) and cross domain cases (dialog than spans multiple domains). It has an average dialog distance length of 3.94 which is roughly 2 user turns and 2 system turns. The existing CC dataset has a Train/Dev/Test split size of 1,280,000/158,043/158,000 respectively.

A.7 Experimental Setup

We set the Context Carryover framework to the settings that are similar to the current commercial settings and run our experiments. The results in Table 1 and Table 2 use a context carryover threshold of 0.5. The context carryover threshold determines the probability threshold above which the slot will be labeled as a carryover instance (i.e., label of 1). We get the pretrained CLIP embeddings from the open-source CLIP (Radford et al., 2021) repo under the MIT license. For the CC model we use an embedding size of 300 for the dialog encoder and intent encoder. For the slot encoder, CLIP visual and CLP text we use an embedding size of 512. We use CLIP (ViT-B/32) (Dosovitskiy et al., 2020) as the vision encoder and a transformer (Vaswani et al., 2017) based text encoder as described in (Radford et al., 2019) for the CLIP text encoder. For the decoder, we use a single layer transformer based decoder with 12 attention heads which are

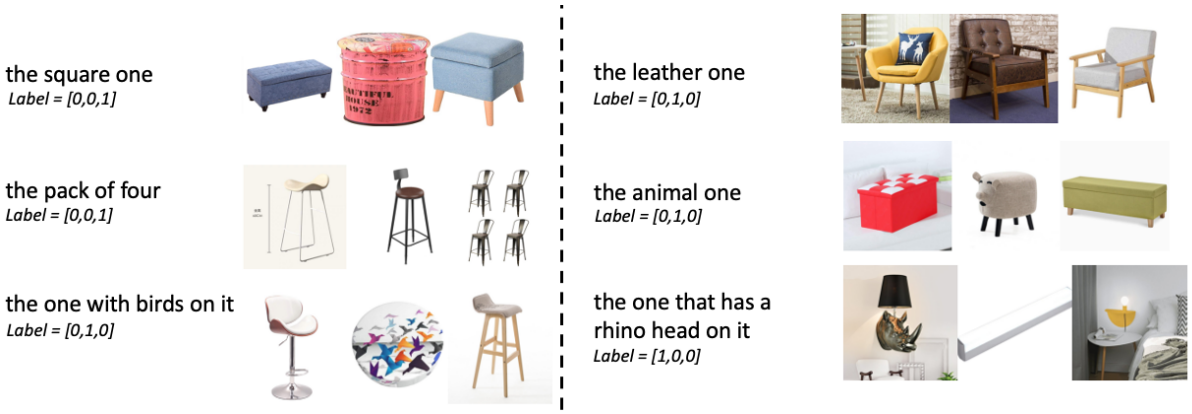

Figure 6: Randomly sampled instances from the MTurk dataset. Each instance is a single referring utterance and multiple associated product images. The label array signifies the ground truth label associated with the referring utterance. A label of 1 signifies the ground truth true label, and 0 otherwise.

```
encoded_list_images = clip_encode_list_images(list_images)                    # (batch x num_list_image_candidates x embedding_dim)
encoded_user_utterance_text = clip_encode_user_utterance(user_utterance_text).  # (batch x 1 x embedding_dim)

normalized_list_images = normalize(encoded_list_images)
normalized_user_utterance_text = normalize(encoded_user_utterance_text)

similarity = dot.product(normalized_list_images, normalized_user_utterance_text)  # (batch x num_list_image_candidates x 1)
```

Figure 7: Pseudocode for vision-text embedding dot product similarity

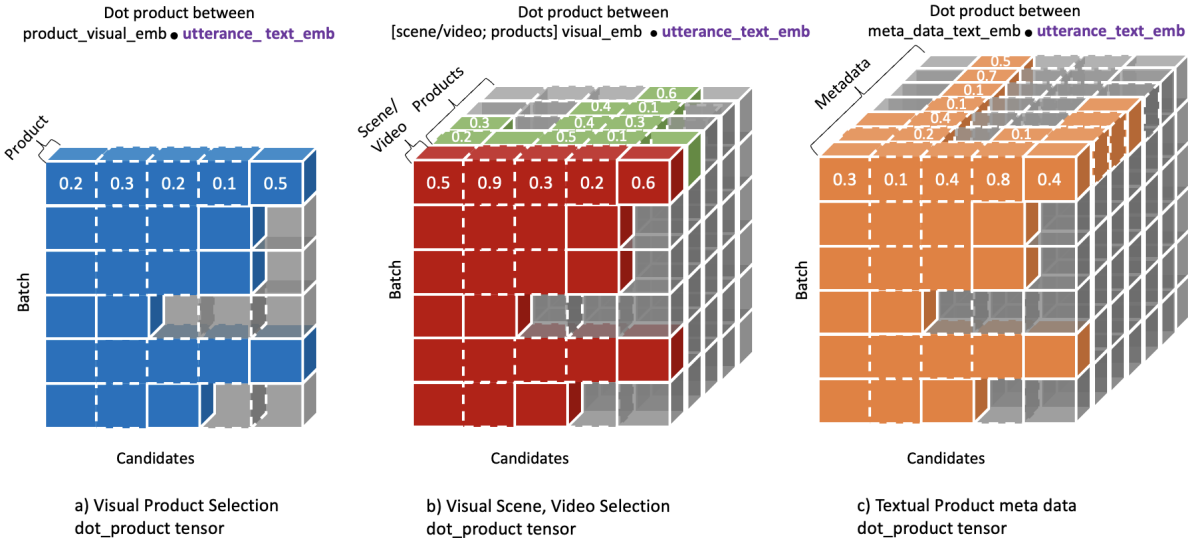

Figure 8: Tensors that contain the dot product between the user's referring utterance text and a) a candidate's single product visual embedding for the "Visual product selection" use case, b) a candidate's scene or video and multiple product visual embeddings for the "Visual scene and video selection" use case, c) a candidate's metadata text embeddings for both the use cases. Here, the metadata are the textual information that are associated with commercial products provided by sellers, marketplace annotators and at times generated by the system. The "candidates" here refer to the visual list item candidates.

then passed to a single layer feed-forward network to make binary decisions over the slots. The model is trained for 100 epochs using a batch size of 160 with an Adam optimizer and learning rate of 0.001. We train on a single p3.16xlarge instance, which consists of 8 GPUs.

A.8 ALBEF results

We also experiment with ALBEF (Li et al., 2021b) embeddings trained using a Large Language Model training framework (FitzGerald et al., 2022) and find them to have a similar performance to CLIP as seen in Table 3.

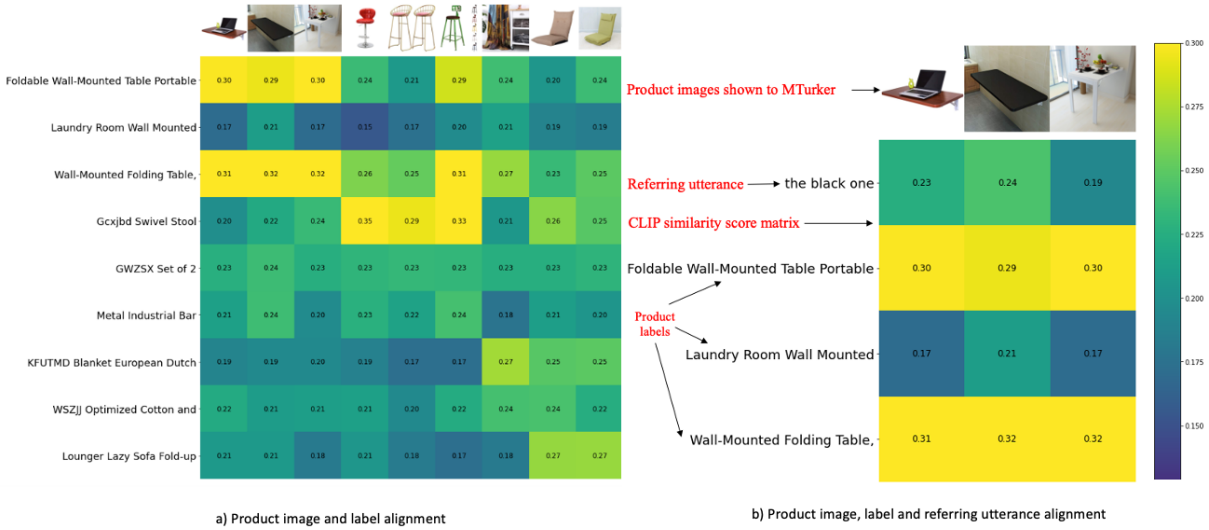

Figure 9: CLIP image-text alignment between images, labels, and referring utterance.

Table 3: Comparison with CLIP and ALBEF

#	Visual Emb.	Model	Slot Level					
			Weighted (P, R, F1)				Acc.	Δ
			P	R	F1	Δ		
1	None	Baseline	0.59	0.63	0.60	-	0.6301	-
2	CLIP	+ Visual	0.78	0.78	0.78	31.12%	0.7811	23.96%
3		+ {Visual + Utterance Text + Similarity}	0.85	0.85	0.85	42.15%	0.8484	34.65%
4	ALBEF	+ Visual	0.77	0.78	0.77	29.97%	0.7772	23.35%
5		**+ {Visual + Utterance Text + Similarity}**	**0.86**	**0.86**	**0.86**	**44.34%**	**0.8617**	**36.76%**