Multimodal Coreference Resolution

IEEE Publication Technology, *Staff, IEEE,*

***Abstract*—In a situated dialog system, multimodal coreference resolution is the task of identifying which entity the user is referring to within a certain context defined by both natural language and visual modalities. This is a crucial task to tackle in order to build effective task-oriented dialog systems. In this work we review the state-of-the-art multimodal coreference resolution models that are mainly based on large language models. One subtrack of the DSTC10 Challenge focused on assessing multimodal coreference resolution using the SIMMC2 dataset. We study and replicate the best proposed systems and we analyze their strengths and weaknesses. We carry out a set of experiments to assess the models' adaptability to different scenarios they were not trained on, showing what makes a model more domain independent. We also propose several modifications that boost the overall performance by including object descriptions as part of the models' input. Moreover, an auxiliary task head at the output of the models is attached to predict the number of referred objects in the last user utterance and effectively refine the final predictions. The most common errors of the models are analyzed as well and they are the inspiration for some of the proposed modifications. Finally, the human-level performance is estimated using a random subset of the SIMMC2 set. We show that the models after the proposed improvements of this project are performing at the human-level.**

***Index Terms*—Multimodal Coreference Resolution, DSTC10, SIMMC2.0, Transformers, BART, UNITER.**

I. Introduction

LANGUAGE and how it can be interpreted by machines has always been a hot topic in machine learning. Large language models have been proven to tackle many dialog-focused tasks effectively. One crucial task to implement leading-edge task-oriented dialog systems is multimodal coreference resolution.

In a linguistic context, *coreference resolution* is defined as the task of determining expressions, such as noun phrases and pronouns, which refer to the same real-word entity. Resolving coreferences makes sentences become self-contained and potentially allows language models to understand more easily their meaning within the dialog. There are many pieces of work in the field of Natural Language Processing (NLP) related to coreference resolution: for example [3] tackles coreference resolution of noun phrases in unrestricted text, and [4] present an approach to pronoun resolution based on syntactic paths.

Similarly, the term *multimodal coreference resolution* is defined as the task of identifying which word or expression co-refer to the same entity in an image or video. This is a crucial problem since many visual agents have to link coreferences (e.g. pronouns) to their corresponding general reference (e.g. nouns or noun phrases) and only then solve their main task, such as grounding [5], [6] or visual question answering [7]. Like many other challenging problems, *multimodal coreference resolution* (MMCR) requires the processing of both natural language and visual features. In a situated dialog system, multimodal coreference resolution focuses on identifying which object the user is referring to within a certain context. This context is defined using both natural language (dialogs or object descriptions) or visual (images or scenes) modalities.

This project has the goal to review existing state-of-the-art solutions that tackle the coreference resolution task, analyze their strengths and weaknesses and propose several improvements to boost the performance. In particular, Section II introduces the DSTC10 competition, Section III presents the considered SIMMC2 dataset, Section IV describes the baselines methods and the proposed modifications. All the experiments and results are shown in Section V, and we conclude in Section VII summarizing and highlighting the main contributions of this project.

II. Related Work

The tenth *Dialog System Technology Challenge* (DSTC10) is the 10th version of a set of dialog related challenges that aims to encourage the community to build and boost existing systems that tackle dialog tasks. Situated multimodal conversational agents, such as virtual assistants that are able to interact with humans, are considered essential and they are focused on the second track for the 2021 competition: SIMMC2.0 (Section III).

The second track of the DSTC10 challenge considers the tasks proposed in [1] on the SIMMC2.0 dataset with the goal of creating and improving immersive multimodal conversational agents. The investigation of this project is also based on the SIMMC2.0 dataset, that is the second version of a task-oriented dialog dataset for situated and interactive multimodal conversations published by Meta Research [1]. The first version, SIMMC [2], established the foundations for the real-world assistant agents that can handle multimodal inputs, and perform multimodal actions. For this second version, SIMMC2.0, the corpus is closer the the real word and designed to include realistic dialogs for fashion or furniture shopping scenarios, and situated multimodal user context in the form of co-observed image or virtual reality (VR) environment.

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

[1]Dataset available at https://github.com/facebookresearch/simmc2

III. DATA

The SIMMC2 dataset [1] contains about 11K task-oriented dialogs (around 117K utterances) between a virtual shop assistant and an user taking place in different commercial stores described by scene images and item descriptions. The examples are set up within the fashion and furniture shopping domains. There are about 7.2K dialogs from fashion domain and 4K from furniture. The fashion domain is expected to be harder to successfully assess since there are more items per scene and they might be visually overlapped when hanging on walls or on shelves. The Table I describes dialog and scene statistics of SIMMC2.0 dataset.

Total no. dialogs	11244
Total no. utterances	117,236
No. dialogs from fashion domain	$\sim$ 7.2K
No. dialogs from furniture domain	$\sim$ 4K
Total no. scenes snapshots	1566
Average no. words per user turns	12
Average no. words per assistant turns	13.7
Average no. utterances per dialog	10.4
Average no. objects mentioned per dialog	4.7
Average no. objects in scene per dialog	19.7

TABLE I
SIMMC 2.0 DATASET STATISTICS

The dialogs of SIMMC2.0 dataset were generated through a two-stage data collection pipeline (Fig. 1). In the first step, a scene generator and a dialog simulator produce a situated 3D environment and a assistant-user dialog respectively. The dialog simulator randomly picks an agenda for each dialog (request an item, get information, etc.), and then the user and the assistant carry on a conversation until the goal in the agenda is successfully met, or when the dialog reaches the maximum number of turns. In the second phase, human annotators paraphrase the dialog flow to mimic a closer-to-real-world shopping conversation.

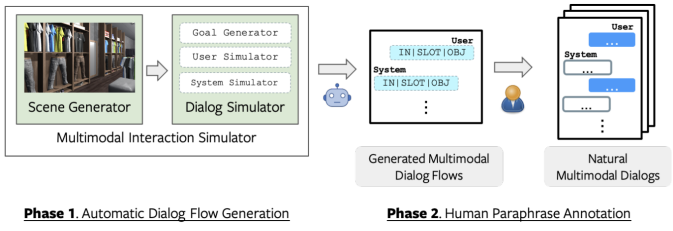

Fig. 1. SIMMC2 two-phase data collection pipeline. In phase 1 a situated multimodal dialog simulator generates the visual environment and the dialog. In phase 2 the dialogs flows are paraphrased by human annotators accordingly a real word application. Representation from [1].

Each dialog is represented by a dialog id, a list of scenes and a list of *turns*. Each turn is defined by a turn id, the assistant utterance, the assistant turn annotations (including belief state and referred objects), the user utterance and the user turn annotations (similar as assistant annotations).

In addition, the scene images are also annotated. For each scene there is a full list of objects appearing in it. These scene objects are described by an object id, the bounding box of the object in the scene image and the 3D position of the item. The dataset also provides a description of each individual object in the form of list of attributes: asset type, type, customer review, color, pattern, brand, sleeve length, price, size and available sizes for the fashion domain; and color, brand, price, type, materials and customer rating for the furniture domain.

The dataset is randomly divided into 4 sets: training set, dev set, dev-test set and std-test set. The proportion, number of dialogs and intention of each set is shown in Table II. This partition is going to be used for the majority of experiments in this project for a fair comparison with other state-of-the-art solutions.

Set	Proportion	No. dialogs	Description
Train	64%	7307	Training set for modeling.
Dev	5%	563	Used for hyperparameter selection and other modeling choices.
Dev-test	15%	1687	Publicly available test set to measure models performance and report results.
Std-test	15%	1287	Left as a held-out hidden set for performing a fair comparison of models.

TABLE II
SIMMC 2.0 SUB-SETS STATISTICS

SIMMC2.0 aims to help the community to build effective multimodal task assistants. To do so, Meta Research proposes four main benchmark tasks: Multimodal Disambiguation (MM-Disamb), Multimodal Coreference Resolution (MM-Coref), Multimodal Dialog State Tracking (MM-DST) and Response Retrieval and Generation (RRG).

IV. METHODS

We focus on two approaches for the investigation that participated in the DSTC10 challenge on the multimodal coreference resolution task. First, the UNITER-based model [9] is studied. It ranked second on this task in the DSTC10 competition and it utilizes rich multimodal inputs to encode the context. Also, we analyze the BART-based model [13] that assesses the four SIMMC2 tasks (MM-Disamb, MM-Coref, MM-DST and RRG) and won the DSTC10 challenge on the multimodal coreference resolution task. In addition, we also propose various modifications aiming to boost the overall performance of the models.

A. UNITER-based model for MM Coreference Resolution

UNITER [10] is a Transformer-based model for image and text universal embeddings and it is the core of the model proposed in [9] to tackle coreference resolution. Given dialog history U, object embeddings $O = o_1 o_2 \dots o_I$ and scene embeddings $S = s_1 s_2 \dots s_J$, the UNITER-based model [9] aims to predict binary object mention labels $Y = y_1 y_2 \dots y_I$ indicating whether each object o_i is referred in the current user utterance.

The object embeddings are computed using the following preprocessed multimodal features: object scene-level index, cropped scene image, 3D object coordinates, non-visual knowledge base (KB), a binary feature `scene_active` that indicates whether an object is in the current active scene; and another binary feature, `prev_mentioned` that indicates whether an item has been previously mentioned in the dialog.

All this preprocessed features are finally fed into a dense layer to compute the object embedding.

Scene embeddings $S = s_1 s_2 \dots s_J$ are computed similar to the object embeddings. It is used the scene index, `scene_active` feature indicates whether the scene is active in the current turn and `prev_mentioned` indicates if the scene was seen in past dialog utterances. After preprocessing all these features, they are concatenated and passed through a dense layer.

The dialog history and the other text-like features are encoded using the BERT [11] tokenizer, that translates text into language tokens. The visual information (images) are encoded using the visual pre-trained CLIP model [12]. After computing dialog history U encoding, object embeddings O and scene embeddings S, they are all fed into an implementation of UNITER. The outputs of it corresponding to each object position are passed through a dense layer to generate the output logits Z, which are transformed into probabilities by Sigmoid function $\sigma(Z)$ for binary classification. This procedure is described in Equation 1.

$$H = \text{Encoder}(U, O, S) \;\rightarrow\; Z = \text{Dense}(H) \;\rightarrow\; Y = \sigma(Z). \tag{1}$$

The overview of the whole model is illustrated in Figure 2.

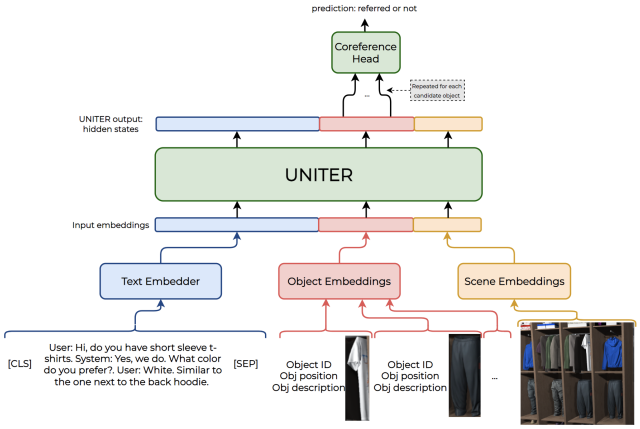

Fig. 2. Overview of the proposed UNITER-based system [9].

B. BART-based model for MM Coreference Resolution

The winner system of the DSTC10 challenge is a BART-based model [13] that assesses coreference resolution among three other tasks. The original team hypothesizes that the model can benefit from solving all tasks at once because the latent representations of the multimodal input features from one subtask are helpful for the other subtasks.

Although the SIMMC2.0 dataset contains images, this system does not use them. Instead, objects and their relations are represented with natural language (tokens). For all subtasks, the input is defined as the concatenation $x := [H_T; U_T; S_{t \leq T}]$ with separators. H_T is the dialog history up to 2 turns to limit the length of the input, i.e, H_T is the set $\{U_{T_2}, A_{T_2}, M_{T_2}, U_{T_1}, A_{T_1}, M_{T_1}\}$, where U_t is the user utterance at time t, A_t is the system utterance at time t and M_t is the multimodal context, that is a set of object indices mentioned by the system at time t. S_t contains the corresponding attributes and locations of all the objects in a scene at time t. Two types of object indices are used: canonical object IDs, that are scene-level IDs and they provide a relational context of the objects within the scene; and global object IDs that intend to capture visual and non-visual attributes of each object.

Additionally, the object locations allow the model infer spatial relations among objects within the scene. 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, the object location is encoded as the following 5-dimensional 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 to a location embedding layer (a fully-connected layer followed by layer norm) to be added with the canonical object ID token encoding.

Even though BART is composed of an encoder and a decoder, the MM-Coref task just uses the encoder to identify the referred objects within the scene. The concatenated canonical object ID (scene-level ID) and the global object ID (unique ID) for each scene object is passed through BART encoder and its output is fed into a binary classification head. This classification head will predict true if that objects is referred to in the current user utterance and false otherwise. The overview of the model focused on the multimodal coreference resolution task is shown in Figure 3.

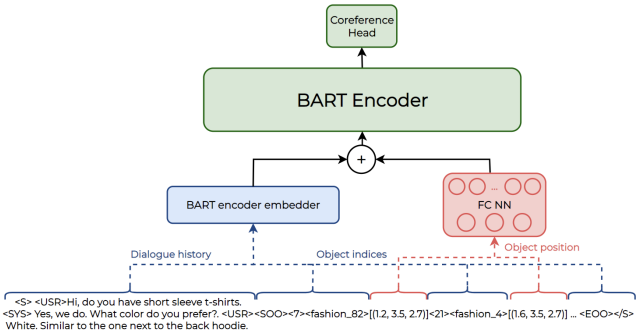

Fig. 3. Overview of the proposed BART-based system [13].

C. Proposed modifications

1) Removing object IDs from UNITER-based model: The first investigated model in the original UNITER's paper [9] was not using object IDs initially. After including them the performance slightly dropped. We believe they are not necessary for coreference resolution and they have to be avoided if we want to identify objects in completely different scenarios, such as different set of objects or domains.

2) Including object attributes in the BART's input: The SIMMC2.0 dataset provides `metadata` files with descriptions of all the objects as a list of non-visual and visual attributes. However, only non-visual attributes can be used at inference time in the DSTC10 competition, since the visual attributes are expected to be recognized by a descriptor system. The BART-based model incorporates both visual and non-visual attributes to train the BART encoder embedder. The BART-based model applies the technique proposed in [12] which intends to encode the global object IDs close to each

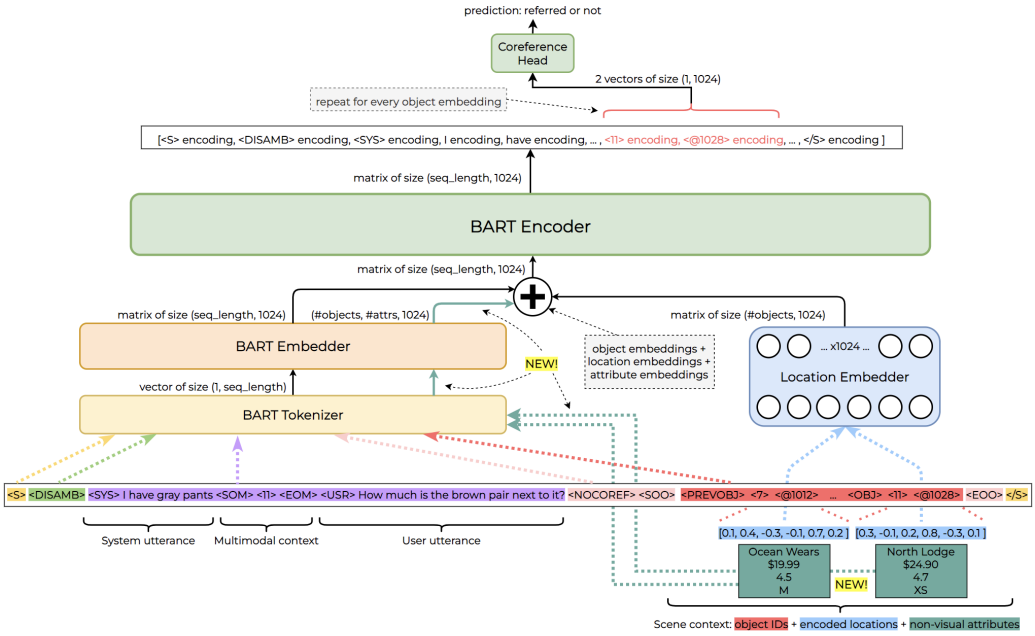

Fig. 4. Including textual attributes in the BART-based model input as they were natural language descriptions of the objects. Each attribute is tokenized by the BART tokenizer and then their embeddings are calculated. The attribute embeddings are added to the corresponding canonical object ID embeddings and to the location embeddings before being fed to the BART Encoder. Finally, the encodings corresponding to each object canonical ID and global ID are used as input for the coreference head, that predicts if the object was referred or not.

corresponding attributes in the high-dimensional embedding space. Then, the model can recognize the object attributes by just the global ID at inference time. We believe that including non-visual attributes as they were object descriptions in the BART-based model's input (Fig. 4), similar to the object IDs and object location encodings, can provide additional signal to link an object to its textual description. We also consider using visual attributes to evaluate what would happen if we had an oracle object recognition system that could provide textual descriptions for each object based on its visual characteristics.

3) Auxiliary task output head: The described systems may not only have problems when identifying the referred object/s, but also they struggle at recognizing the number of referred items in the last user turn. This problem can lead to models that are *under-predicting*, *i.e.*, the number of predicted referred objects is smaller than the true number of referred objects. Aiming to address this issue, the models are extended with an additional auxiliary output head. This head has the goal of predicting the number of referred objects in the last user turn, and this prediction is used at inference time to modify the set of hypothesized referred items and raise the overall performance. We use heuristics to post-process the model's outputs using the number of objects (N) predicted by the auxiliary head. If the number of predicted referred items N is larger than the number of detected referred objects by the coreference head, we force the model to make more predictions by estimating the probabilities of the objects of being referred. These object probabilities are estimated using the output of the coreference head and the *softmax* function. If $N \geq 3$, then only the 3 most probable objects are retrieved, since we consider referring to 4 or more objects in a single utterance is very unlikely and this allows us to formulate a multi-class classification problem with 4 classes.

V. Experiments and Evaluation

We first investigate the effect of removing object IDs from UNITER's object embeddings. The resulting performance is **0.758 F1 score**, increasing the performance by 3%. This means the UNITER-based model is not relying on object IDs to identify the referred items.

Moreover, we evaluate the BART-based model using only the coreference output head. The final performance is **0.748 F1 score**, that shows not only the model is not benefiting from tackling all SIMMC2 tasks at the same time, but it is slightly improving since it can focus on learning the MM-Coref task.

Table III shows the performance of the replicated models and the mentioned improvements and they are compared with the results obtained in the original DSTC10 papers [9], [13].

Model	Description	Original	Ours
GPT-2	SIMMC2.0 Baseline	0.366	0.381
UNITER	Best individual model submitted to DSTC10	0.674	0.675
UNITER	Improvement post-evaluation incorporating previously mentioned objects	0.728	0.726
UNITER	Removing IDs and considering previously mentioned objects	-	**0.758**
BART	4-heads system submitted to DSTC10	**0.743**	0.742
BART	System with just coreference head	-	**0.748**

TABLE III
COMPARISON BETWEEN REPORTED PERFORMANCE IN THE ORIGINAL PAPERS [1], [9], [13] AND OUR REPLICATION RESULTS.

A. Comparison of SIMMC2 domains

The SIMMC2 dialogs are set in two different shopping scenarios: fashion and furniture domains. In total there are around 11K dialogs divided in 7K fashion and 4K furniture dialogs. We want to study how the performance of the UNITER-based and BART-based models vary on each of the domains because this analysis can give insights about how a model can generalize to different domains. Table IV contains the results on each of the SIMMC2 domains.

Model	F1 Score	
	fashion domain	furniture domain
UNITER-based	0.736	0.843
BART-based	0.721	0.860

TABLE IV
PERFORMANCE COMPARISON ON EACH OF THE SIMMC2 DOMAINS.

We found that the performance on the furniture domain is much higher with both models. We hypothesize that the fashion domain is more challenging since the number of objects per scene tends to be larger and the items might be visually overlapped. Plot 5 shows that the average number of objects per furniture scene is slightly over 10, whereas in the fashion domain the mean is above 30, surpassing 55 items in some scenes.

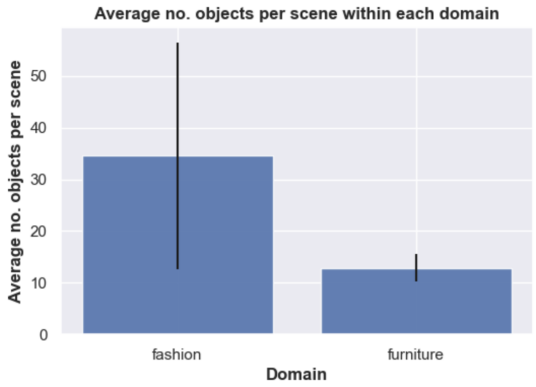

Fig. 5. Average number of object per scene in each of the SIMMC2 domains.

Therefore, the number of candidate objects in the scene and their visual overlapping impact significantly the overall performance of the systems.

B. Visual and non-visual attributes in the BART-based model

The usage of just non-visual attributes is first investigated, which was legal under the DSTC10 competition rules. Figure 4 illustrates the considered overall system. In the proposed approach, textual object attributes are also passed through the BART tokenizer and the BART embedder to obtain their embeddings. Then, these embeddings are added to object location embeddings and the corresponding object IDs embeddings as in the original model. A version of this model is trained using the non-visual attributes of each object as part of the input. The non-visual attributes in the fashion domain are *brand*, *price*, *size* and *customer rating*. In the furniture domain they are the same but *size* in this case is replaced by *material*. After training for 10 epochs the final performance is **0.760 F1 score** on the `devtest` set, that means an absolute performance increase of more than 1%, as reflected in Table V.

Although visual attributes from the `metadata` files are banned in the DSTC10 challenge, we analyze the benefits of incorporating that piece of information. Visual attributes are included along with non-visual attributes as part of the input of the BART-based model. In particular, the considered visual attributes of the fashion domain are *color*, *type*, *asset type* and *pattern*, whereas in the furniture domain they are *color* and *type*. After 10 epochs of training the BART-based model using both non-visual and visual attributes in the input achieves **0.771 F1 score**, increasing the absolute performance by another 1%, which is shown in Table V.

Base model	non-visual attrs.	visual attrs.	Legal on DSTC10	F1 Score
BART†			✓	0.748
BART†	✓		✓	0.760
BART†	✓	✓		0.771

TABLE V
EFFECT OF INCLUDING NON-VISUAL OR VISUAL ATTRIBUTES AS PART OF THE INPUT IN THE BART-BASED MODEL. THE FOLLOWED APPROACH IS DESCRIBED IN DIAGRAM 4. BART† CORRESPONDS TO THE BART MODEL THAT USES ONLY THE COREFERENCE HEAD.

To sum up, we have empirically proven that incorporating textual object descriptions as part of the input boosts the performance of the overall model from 0.748 to over 0.76 and 0.77. Using visual attributes is not valid in the DSTC10 competition, but we show what would happen if we remove the overhead of a possible object descriptor system.

C. Domain and scene generalization of the models

We assess the adaptability of the models to different scenarios. A possible deployed virtual assistant might need to operate in a different domain than the one it was trained on. Moreover, although the final environment is within a known domain, the objects can be unseen during training or even not stored in the database, then prior information about their non-visual and visual attributes would not be available at inference time. Therefore, we study the models behavior on unseen objects (out-of-domain) and on unseen scenes (different physical stores). This investigation requires to re-arrange the dialogs of the SIMMC2 dataset, so we can evaluate the models in different domains or in unseen scenes. Table VI describes all the subsets considered for the cross-domain investigation. Some sets are used for training (**FASH-14K** and **FURN-12K**), and each of them contain dialogs from just one domain. The SIMMC2 `devtest` set is divided into **FASH-6K** and **FURN-2K** depending on the domain of each dialog. Additionally, another three subsets are created: **ID** (in-domain) is used for testing and contains dialogs with the same distribution as **FASH-14K**. The **IDHO** (in-domain held-out) set also contains dialogs from the fashion domain, but only from the `cloth_store_1498649_woman` shop, so its scenes are unseen at test time since the physical building is different compared to the ones in **FASH-14K** training set. Finally, **OOD** (out-of-domain) uniquely contains dialogs from the furniture domain.

Name	No. examples	Description
FASH-14K	$\sim$ 14.6K	Set used for training. All examples are within fashion domain and they are not set in `cloth_store_1498649_woman` shop.
FURN-12K	$\sim$ 12K	Set used for training. All SIMMC2.0 furniture domain examples.
FASH-6K	$\sim$ 6.5K	Set of fashion domain examples of the SIMMC2.0 `devtest` set.
FURN-2K	$\sim$ 2K	Set of furniture domain examples of the SIMMC2.0 `devtest` set.
ID	$\sim$ 9K	Set used for testing. The samples were randomly picked from fashion domain and not placed in `cloth_store_1498649_woman` shop. The distribution of this test set is expected to match `train-cd` distribution.
IDHO	$\sim$ 9.5K	Set used for testing. All conversations are set in `cloth_store_1498649_woman` shop. The object distribution and the locations may vary from `train-cd` dataset.
OOD	$\sim$ 9.5K	Used for testing. All examples are within furniture domain. It evaluates the adaptability of the models to different domains.

TABLE VI
DIVISION OF THE SIMMC2 DATASET FOR CROSS-DOMAIN EVALUATION.

First, we evaluate the performance of the models on unseen domains. A multimodal conversational system deployed on an unseen domain can rely on generic visual (extracted from the image) and non-visual (extracted from meta-information) attributes. Table VII shows the results of the models trained on out-of-domain datasets along with the results of the models trained on the standard SIMMC2 training set (in-domain). When a result is specified as 'out-of-domain' (3rd column) it means the model has been trained on **FASH-14K** and tested on **FURN-2K**, or trained on **FURN-12K** and tested on **FASH-6K**, accordingly to the 4th and 5th columns that report the result on each test set.

Base model	Attrs.	Training set specification	F1 Score on FASH-6K	F1 Score on FURN-2K
UNITER†	NV	in-domain	0.736	0.843
UNITER†	NV	out-of-domain	**0.425**	**0.525**
BART†		in-domain	0.721	0.860
BART†	NV	in-domain	0.731	0.861
BART†	NV+V	in-domain	**0.743**	**0.868**
BART†		out-of-domain	0.200	0.431
BART†	NV	out-of-domain	0.194	0.457
BART†	NV+V	out-of-domain	0.210	0.516

TABLE VII
COMPARING MODELS TESTED ON A KNOWN DOMAIN VS A DOMAIN NOT SEEN AT TRAINING. THE MODELS WERE TRAINED ON THE SIMMC2.0 TRAINING SET (IN-DOMAIN) AND ON A SUBSET OF EXAMPLES BELONGING TO A DIFFERENT DOMAIN THAN THEY ARE EVALUATED (OUT-OF-DOMAIN). FOR A OUT-OF-DOMAIN TRAINING SET, IT IS BUILT WITH FASHION EXAMPLES IF IT IS EVALUATED ON **FURN-2K**, AND ANALOGOUSLY, IT IS BUILT WITH FURNITURE EXAMPLES IF IT IS TESTED ON **FASH-6K**. NV AND V STAND FOR NON-VISUAL AND VISUAL ATTRIBUTES RESPECTIVELY.

The UNITER-based model performs better than the BART-based on OOD scenarios. This is because the UNITER-based model is not relying on memorizing the trained objects and it is extracting more general features from the objects, scenes and the dialog context. Last column informs UNITER-based models partially trained on the furniture domain effectively resolve coreferences with about 0.84 F1 score, and BART-based models around 0.86. However, BART-based models suffer more degradation when tested on a OOD set. The drop in performance of the UNITER-based models is around 30%, whether the BART-based models decrease more than 40% F1 since they heavily rely on object encodings modeled at training time. Introducing visual attributes improves the results of the BART-based models, achieving a performance around 0.51, close to the UNITER-based models tested on OOD. This is because the visual attributes try to alleviate the lack of training of those new objects. Similarly with the 4-th column, BART-based models perform better on a in-domain test set, but they struggle more than UNITER-based models when evaluated on an OOD test set, in this case, a fashion test set. UNITER models drop again about 30% F1 score, and BART-based system suffer even more degradation. Visual attributes help, but this time they are not enough to reach UNITER's OOD performance. Definitely, BART-based models cannot be used in OOD scenarios. The UNITER-based models are superior in this aspect, but they are not performing extraordinary well either.

We also investigate how the models perform on unknown stores (not seen at training time) and different object distributions. We employ **FASH-14K** set for training, and **ID**, **IDHO** and **OOD** sets for testing. Objects in a certain store can be distributed differently than in other buildings although they are all within the same domain. The **IDHO** set assesses the model performance in seen domains but with different object distributions. The results are shown in Table VIII. The BART-based models perform better on **ID** and on **IDHO** than the UNITER-based model. BART-based models are performing considerably well when the possible objects were seen at training time. Comparing the performance between **ID** and on **IDHO** test sets, UNITER-based models degrade by 7% when the setting of the stores changes or when the objects frequency is altered compared to the training set. In contrast, BART-based systems are robust if they were trained on the same set of objects, although their frequencies are different. We hypothesize this is because the UNITER-based models rely more on the visual features, such as images, scenes or object coordinates, whether the BART-based models can encode them using the global IDs embeddings. However, if the models are tested on an unseen set of items (**OOD** set), the UNITER-based models are performing about 8% better since they have introduced some generalization by not relying on object IDs. The fact that BART-based models are performing better on **IDHO** set than on **ID** set is because the typical settings of the scenes in **IDHO** dataset are easier to analyze, since they have fewer objects as shown in Plot 6.

Base model	Attributes	ID	IDHO	OOD
UNITER†	NV	0.694	0.621	**0.549**
BART†		0.712	0.744	0.456
BART†	NV	0.675	0.740	0.373
BART†	NV+V	**0.718**	**0.744**	0.451

TABLE VIII
PERFORMANCE OF THE UNITER AND BART-BASED MODELS IN DATASETS WITH DIFFERENT SCENE AND OBJECT DISTRIBUTIONS.

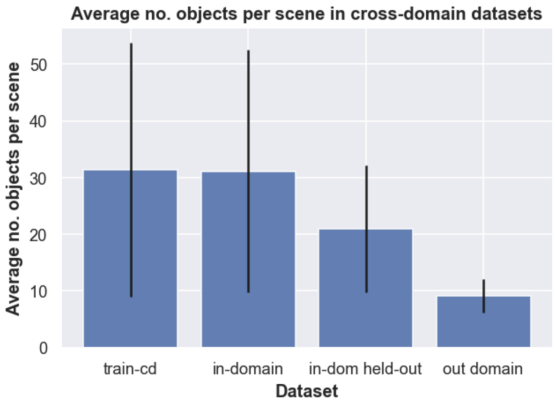

Fig. 6. Average no. of objects per scene in each of the cross-domain datasets.

D. Effect of the reference type

We study the impact of mentioning the target object in the conversation before the last turn. Depending on the reference type, an target object could have been previously mentioned in the dialog, so there is more information in the conversation context. In contrast, some objects are referred just in the last user turn (new), so they require visual feature processing to be identified.

Table IX shows the main reported results so far split depending on the reference type of the target item. The performance is measured on the `devtest` set, where the 57% of targets were contained in the conversation context, i.e., they were previously mentioned in the dialog. The rest (43%) were new objects recently referred in the last user turn. UNITER and BART overall performance is between 0.75 and 0.77 F1 score. UNITER has significantly higher performance on *mentioned* with **0.837 F1** while BART achieves higher performance on *new* objects (between 0.71 and **0.73 F1**). This indicates that BART's encoding of the visual information using text is more efficient than UNITER's ability to process visual features from an image.

Base model	IDs	Attrs.	F1 score mentioned	F1 score new objs.	F1 score overall
UNITER†	✓	NV	0.821	0.594	0.726
UNITER†		NV	**0.837**	0.644	0.758
BART†	✓		0.783	0.712	0.748
BART†	✓	NV	0.796	0.722	0.760
BART†	✓	NV+V	0.807	**0.733**	**0.771**

TABLE IX
PERFORMANCE OF UNITER AND BART-BASED MODELS DEPENDING IF THE TARGET OBJECT WAS PREVIOUSLY MENTIONED IN THE DIALOG OR NOT (NEW OBJECT).

We show that UNITER-based models are better on mentioned items, and the BART-based models are superior identifying new objects. In section V-F we use this finding to combine these models and boost the overall performance.

E. Addressing under-prediction

We implement the auxiliary task output head in both UNITER and BART-based models. In both cases, at training time, the total loss of the model is the weighted sum of the coreference head loss and in the auxiliary head loss, as formulated in Equation (2).

$$\mathcal{L}_{\text{total}} = \lambda_{\text{mm-coref}}\mathcal{L}_{\text{mm-coref}} + \lambda_{\text{n-targets}}\mathcal{L}_{\text{n-targets}}. \quad (2)$$

$\mathcal{L}_{\text{n-targets}}$ is the loss of the new auxiliary head that predicts the number of target objects, and $\lambda_{\text{n-targets}}$ is the weight assigned to it which value has to be tuned next to the value of $\lambda_{\text{mm-coref}}$. Both heads are trained at the same time using *cross entropy loss* and only at inference time the predictions of the auxiliary head are employed to modify the coreference head predictions. We use heuristics to post-process the model's outputs using the number of objects (N) predicted by the auxiliary head to address 'under-prediction'.

We compare the effect of adding the new auxiliary task head to the studied systems and the results are shown in Table X. After tuning the hyperparameters $\lambda_{\text{n-targets}}$ and $\lambda_{\text{mm-coref}}$ of Equation (2), all the models improved with respect its equivalent version without the additional task head.

Base model	Attrs.	Aux. head	F1 score mentioned	F1 score new objs.	F1 score overall
UNITER†	NV		0.837	0.644	0.758
UNITER†	NV	✓	**0.844**	0.644	**0.761**
BART†			0.783	0.712	0.748
BART†		✓	0.812	0.693	0.752
BART†	NV		0.796	0.722	0.760
BART†	NV	✓	0.827	0.700	0.763
BART†	NV+V		0.807	**0.733**	0.771
BART†	NV+V	✓	0.835	0.715	**0.775**

TABLE X
PERFORMANCE COMPARISON OF UNITER AND BART-BASED MODELS AFTER INCORPORATING THE NEW AUXILIARY TASK HEAD. THE RESULTS ALSO REPORT THE PERFORMANCE ON PREVIOUSLY MENTIONED ITEMS IN THE DIALOG OR NOT (NEW). BOTH UNITER† AND BART† ARE THE BEST DSTC10 SUBMISSIONS ADDING MULTIMODAL CONTEXT (PREV. OBJS.) AS PART OF THE INPUT. UNITER† IS NOT USING OBJECT IDS.

The UNITER-based model increased the performance up to 0.05, surpassing **0.760 overall F1 score** mark. Similarly, all BART-based systems increased the overall performance as well. The best model is the BART-based system that uses non-visual (NV) and visual (V) attributes as part of the input and the new auxiliary task head, reaching **0.775 F1 score**, that is a ∼0.05 increase compared to its version without the extra head. Remember that this version of the BART-based model is not valid at DSTC10 competition since it is using visual attributes at inference time. However, the model using just non-visual attributes would be legal, and it is achieving **0.763 F1 score**. The vanilla version is also enhanced after incorporating the new head.

In addition, we report the results on the objects previously mentioned and new ones. Most of the 'mentioned' objects

can be resolved using natural language context while 'new' objects require processing of visual features. As expected, we observe that the systems usually perform better on mentioned objects than in objects recently referred in the last user turn. With the proposed auxiliary head, both systems improve the performance on mentioned (UNITER-based improved by 1% and BART-based went up by 3%) but not on new objects. This can be caused by the fact that reference probabilities on new objects are less accurate because they require processing of visual information. Hence applying the proposed heuristics does not help the performance on new objects.

F. Model combination

We observe in Tables X and IX that UNITER-based models are better at identifying previously mentioned objects in the dialog, and BART-based systems excel at resolving recently referred items. Therefore, we propose to combine the strengths of the two approaches: the UNITER-based model would aim to identify the referred objects that were previously mentioned, i.e., the objects within the multimodal context; and the BART-based model would focus on objects not present in the multimodal context.

Table XI shows the results after combining the models. The best performing model combination achieves **0.806 F1 score**. However, this model uses textual visual features that were not permitted in the DSTC10 competition. The system that would be legal in the challenge achieves **0.800 F1 score**. The auxiliary head is useful with the UNITER-based model because it is enhancing the model on mentioned objects, but it is not helpful with the BART-based model, since this head was not improving on 'new' items.

UNITER model config.			BART model config.			F1 Score
IDs	Attrs.	Aux. head	IDs	Attrs.	Aux. head	overall
	NV	✓	✓	NV	✓	0.789
	NV	✓	✓	NV		**0.800**
	NV	✓	✓	NV+V	✓	0.797*
	NV	✓	✓	NV+V		**0.806***

TABLE XI
RESULTS AFTER COMBINING UNITER AND BART BASED MODELS. RESULTS MARKED WITH '*' WOULD NOT BE VALID IN DSTC10 BECAUSE OF THE USAGE OF TEXTUAL VISUAL ATTRS.

To sum up, we could combine the strengths of both models and achieve **0.800 F1 score** on the DSTC10 competition, or even **0.806 F1 score** in a real scenario with textual visual attributes available at inference time.

G. Estimating human-level performance

On a lot of machine learning tasks, systems rapidly become better at the beginning, but they slow down usually when *human-level performance* is reached. We sample 100 random examples from the SIMMC2 `devtest` set to evaluate human performance on this task and estimate a hypothetical upper bound for the models' performance. Each example consists of the dialog history between the user and the assistant, the multimodal context (IDs of the mentioned objects), the current scene image and all the IDs of the objects appearing in the scene.

Three human annotators were asked to complete the task of identifying the referred objects in the last user utterance reporting their IDs. They annotators could see the same dialog context as the models were using, the IDs of the mentioned objects (multimodal context), and the scene image including all objects' IDs. The average human-level performance is **0.822** $\pm$ **0.025** F1 score on this random subset of the `devtest` set as illustrated in Table XII.

Human annotator	F1 score on mentioned	F1 score on new objs.	F1 score overall
no. 1	0.906	0.703	0.857
no. 2	0.872	0.571	0.803
no. 3	0.881	0.556	0.805
Average	0.886	0.610	0.822

TABLE XII
RESULTS OF EVALUATING HUMAN ANNOTATORS ON A RANDOM SUBSET OF 100 EXAMPLES OF THE DEVTEST SET.

We observe the annotators tend to make similar mistakes and they often struggle with the same examples. The detailed analysis of systems and human errors is described in Section VI. We compute the Inter-Annotator Agreement by computing the Cohen's Kappa estimator [14] using Equation (3).

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \tag{3}$$

where P_o is the relative observed agreement among annotators, and P_e is the hypothetical probability of chance agreement.

Table XIII shows that the estimated Cohen's κ mean is 0.856, which indicates high agreement between the annotators [14].

Annotators pair	Cohen's κ
no. 1 and no. 2	0.838
no. 1 and no. 3	0.879
no. 2 and no. 3	0.851
Average	0.856

TABLE XIII
ESTIMATING THE INTER-ANNOTATOR AGREEMENT BY THE COHEN'S KAPPA STATISTIC. THE AVERAGE COHEN'S KAPPA IS OVER 0.8, WHICH IS CONSIDERED TO MEAN HIGH ANNOTATION RELIABILITY.

We can calculate the performance of the studied models on this random subset and compare it with the estimated human-level performance. The results are shown in Table XIV, where UNITER† and BART† are the best DSTC10 submissions adding multimodal context (prev. objs.) as part of the input, and UNITER† is not using object IDs.

Surprisingly, the models can not only match human-level performance, but also the best performing models are able to beat it. Note that the scores are higher than in previous sections since the proportion of targets that are previously mentioned in the dialog happened to be higher in this test set. Although human evaluation was performed on a small dataset, the results suggests that the improved models are already achieving human-level performance.

Base model	Attrs.	Aux. head	F1 score mentioned	F1 score new objs.	F1 score overall
UNITER†	NV		0.887	0.706	0.846
UNITER†	NV	✓	0.873	0.632	0.811
BART†			0.860	0.826	0.849
BART†		✓	0.845	0.679	0.784
BART†	NV		0.844	0.776	0.820
BART†	NV	✓	0.887	0.778	0.848
BART†	NV+V		0.860	0.706	0.806
BART†	NV+V	✓	0.905	0.778	0.859
Average human performance			0.886	0.610	0.822

TABLE XIV
RESULTS OF EVALUATING MODELS ON RANDOM SUBSET OF 100 EXAMPLES.

VI. ERROR ANALYSIS

We manually examine errors made by the models and by the human annotators. The following examples illustrate the common errors.

Models are under-predicting: Figure 7 illustrates how the models are not able to recognize that there are some referred items. We believe the models are assigning more likelihood of being referred to those objects, but not enough to reach the threshold value to be considered as a positive prediction. This type of error would be fixed if we could cleverly reduce the prediction threshold in some particular cases. We successfully assessed this type of error using an additional task head and a set of heuristics to post-process the models predictions.

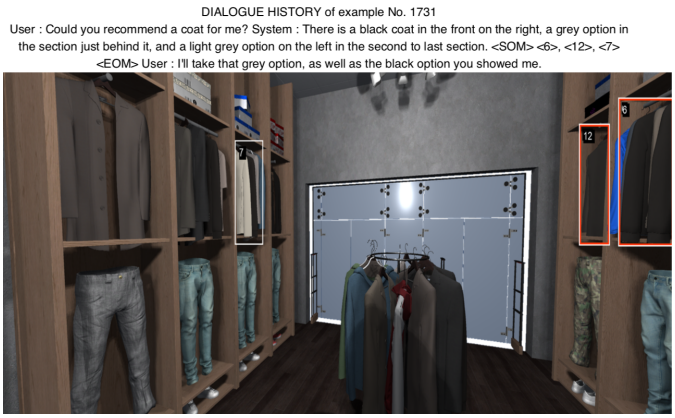

Fig. 7. Two referred coats are not identified, marked in red squares.

Models are over-predicting: The models can predict as referred more items than they are expected. In Figure 8 we can observe that this is a common mistake if there are several objects with the same properties. In the image, several jeans are shown in the store's cubby, but just two of them are the true targets. We hypothesize the models are identifying the objects attributes or descriptions in the dialog and then they are assigning those objects higher probability, without sufficiently taking into account the last utterance and the number of actual referred items.

Models are struggling with locations: The previous example (Fig. 8) also exemplifies that some referring expressions

Fig. 8. All jeans are predicted because they have equal descriptions.

use relative locations to ground the mentioned objects like "two cubbies over", so the objects have to be identified by first knowing what is a "cubby". Pre-trained large language models might be able to know spacial expressions like "on the left", "on top of something", etc. but they still struggle to identify location expressions like "rightmost shelf" or "two cubbies over". Not only the expressions are more complex, but also they are dependent on potentially unknown items, such as a cubby, a rack or a shelf. Large language models probably are not pre-trained on them, and SIMMC2 does not provide descriptions of objects that are not targets.

SIMMC2 has some annotation errors: We have observed that some examples of SIMMC2 dataset are not correctly annotated. This is why models are failing in some cases. Figure 9 showcases a case where the ground-truth targets are missing when they should be present. The user is clearly referring to the area rug, but the annotations do not consider any target.

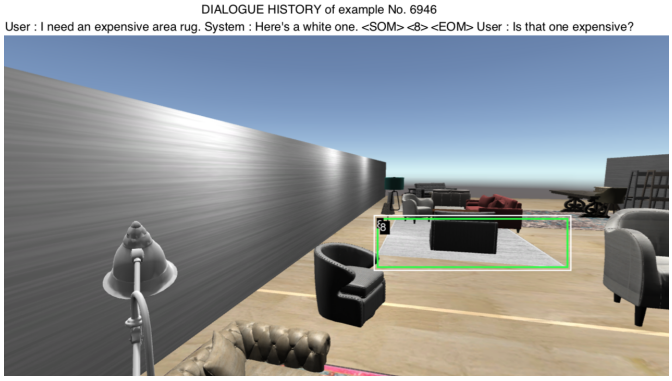

Fig. 9. The example is wrongly annotated: it has no ground-truth targets.

Human annotator errors: The human annotators have an average turn-level error rate of 12%, what means that the human annotators failed on average on 12 examples out of 100. In fact, 7 examples out of 100 were wrongly classified by all the human annotators, and 4 examples out of 100 contained errors on two annotators. So the annotators are struggling mainly in the same subset of examples. The first source of errors comes from the fact that SIMMC2 is not

completely clean, as illustrated in Fig. 9. On the other hand, there are examples that are correctly annotated, but they are just hard to resolve. Figure 10 shows an example where human annotators failed to identify that the referred item was the pair of jeans at the far back of the first cubby. They usually targeted the trousers in the middle or the rightmost part since they are darker and the dialog mentions some "dark blue jeans". Moreover, the other referred object is a grey coat, but there are plenty of items matching that description in the upper shelf.

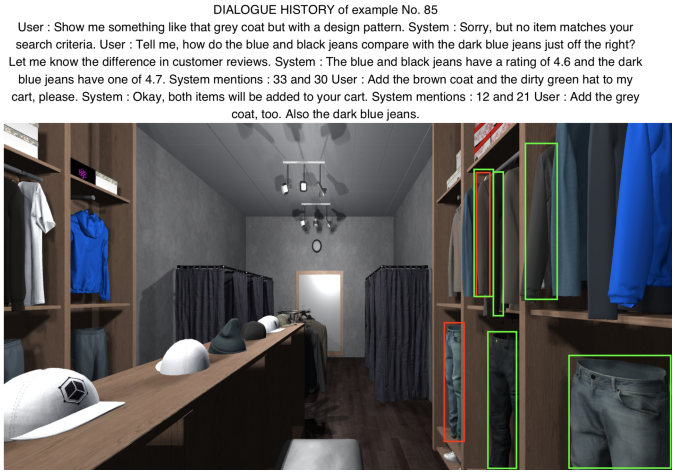

Fig. 10. The annotators fail to identify correctly all referred objects since there are multiple options matching the description.

To sum up, we have seen that several examples are hard to tackle since there are many objects matching the provided description in the conversation. Additionally, the dataset contains incorrectly annotated examples or some ambiguities that can be solved for future versions. However, the models are still struggling with some cases that a human could solve correctly.

VII. Conclusion

We replicate the UNITER-based model [9] and the BART-based model [13] to tackle multimodal coreference resolution task. We propose various improvements, proving that the UNITER-based model does better removing the object IDs from the inputs and that the BART-based model can perform coreference resolution independently of other specific tasks.

The two domains of the SIMMC2 dataset are individually studied, showing that the furniture domain is easier to model since it contains fewer objects in a typical scene.

Moreover, descriptions of the items in the form of list of attributes are shown to be a fruitful source of information for the BART-based model when including as part of the input. We also evaluate the models in unknown scenarios, where the test set contained examples of different nature compared to the ones seen during training. We show that the UNITER-based model can adapt more easily to unseen domains than the BART-based model. The latest performs worse on out-of-domain samples because it heavily relies on object IDs modeled at training time. However, we show that the BART-based model can maintain the same performance although the stores or the object distributions change after training.

Adding a new auxiliary task head at the output of the models is proposed and proven to be beneficial after analyzing the most common errors that the systems were making. We demonstrate the auxiliary task head can provide extra signal at inference time and refine the set of predictions, increasing considerably the overall performance of both models.

We observe that the UNITER-based system is better at recognizing objects present in the conversation context, whether the BART-based model excels at identifying referred objects just in the last user utterance. Therefore, we combine both models to tackle the coreference resolution task using their specific strengths taking into account the objects within the conversation context, reaching 0.800 F1 score.

We compare models performance with the human-level performance on the coreference resolution task on the same dataset. The results exhibit that the models are already performing similar to the human-level.

Finally, an error analysis is carried out to understand the behavior of the models and gain insights to further enhance them. The error analysis can be the starting point of future directions.

References

[1] S. Kottur, S. Moon, A. Geramifard and B. Damavandi. SIMMC 2.0: A task- oriented dialog dataset for immersive multimodal conversations. *Association for Computational Linguistics (ACL)*. 2021.

[2] S. Moon, S. Kottur, P. A. Crook, A. De, S. Poddar, T. Levin, D. Whitney, D. Difranco, A. Beirami, E. Cho, R. Subba and A. Geramifard. Situated and interactive multimodal conversations. *Computing Research Repository (CoRR)*. 2020.

[3] W. M. Soon, H. T. Ng and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Association for Computational Linguistics (ACL)*. 2001.

[4] S. Bergsma and D. Lin. Bootstrapping path-based pronoun resolution. *Association for Computational Linguistics (ACL)*. 2006.

[5] V. Ramanathan, A. Joulin, P. Liang and L. Fei-Fei. Linking people in videos with "their" names using coreference resolution. *European Conference on Computer Vision (ECCV)*. 2014.

[6] S. Kottur, J. M. F. Moura, D. Parikh, D. Batra, and M. Rohrbach. Visual coreference resolution in visual dialog using neural module networks. *European Conference on Computer Vision (ECCV)*. 2018.

[7] P. H. Seo, A. M. Lehrmann, B. Han, and L. Sigal. Visual reference resolution using attention memory for visual dialog. *Neural Information Processing Systems (NIPS)*. 2017.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin. Attention Is All You Need. *Neural Information Processing Systems (NeurIPS)*. 2017.

[9] Y. Huang, Y. Wang, and Y. Tam. UNITER-based Situated Coreference Resolution with rich multimodal input. *Computing Research Repository (CoRR)*. 2021.

[10] Y. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. UNITER: Learning universal image-text representations. *European Conference on Computer Vision (ECCV)*. 2020.

[11] J. Devlin, M. Chang, K. Lee and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *Association for Computational Linguistics (ACL)*. 2019.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever. Learning transferable visual models from natural language supervision. *Proceedings of the 38 th International Conference on Machine Learning*. 2021.

[13] H. Lee, O. J. Kwon, Y. Choi, J. Kim, Y. Lee, R. Han, Y. Kim, M. Park, K. Lee, H. Shin and K. E. Kim. Tackling Situated Multi-Modal Task-Oriented Dialogs with a Single Transformer Model. *Association for Computational Linguistics (ACL)*. 2021.

[14] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*. 1960.