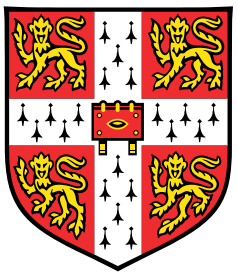
Multimodal Coreference Resolution



Alejandro Santorum Varela

Supervisors: Dr. Svetlana Stoyanchev
Prof. Kate Knill

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Hughes Hall College August 2022

I dedicate this thesis to my beloved parents. They are the reason why I am here today.

Declaration

I, Alejandro Santorum Varela of Hughes Hall College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

The project makes use of the following open source code:

- SIMMC2.0 Dataset and scripts from https://github.com/facebookresearch/simmc2. This is the dataset used for investigation and we also use the provided scripts to replicate the baseline and evaluate the models accordingly DSTC10 Challenge rules.
- UNITER-based model proposed for the DSTC10 Challenge by the New York University of Shanghai, publicly available at: https://github.com/i-need-sleep/mmcoref_cleaned.
- BART-based model proposed for the DSTC10 Challenge and open sourced by the Korea Advanced Institute of Science & Technology (KAIST) AI Laboratory at: https://github.com/KAIST-AILab/DSTC10-SIMMC.
- The PyTorch machine learning library (https://pytorch.org/) for building, training and evaluating the discussed models.
- The NumPy (https://numpy.org/) and Matplotlib (https://matplotlib.org/) libraries for data wrangling and plotting, and the Pillow (https://pypi.org/project/Pillow/) library for image processing.

Word count: 14969

Alejandro Santorum Varela
August 2022

Acknowledgements

I would like to thank my two supervisors, Dr. Svetlana Stoyanchev and Prof. Kate Knill for their countless suggestions, guidance and support throughout the whole project. I am truly grateful for inspiring and teaching me during these last months. I would also like to thank Dr. Simon Keizer and Dr. Rama Doddipatla for their time and insightful discussions at every meeting.

Moreover, I am highly thankful to Toshiba Europe Ltd. for the provided computing resources during the internship, and the Cambridge University Engineering Department for all the given service in the last year.

Finally, I massively appreciate my family and friends for every bit of encouragement that you grant me every single day.

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, also known as Transformers. In fact, one sub-track 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 aiming to improve them.

We carry out a set of experiments to assess the models' adaptability to different domains and scenarios that they were not trained on, showing key factors that make a model more domain independent.

We also propose several modifications that boost the overall performance by including object descriptions in the form of lists of attributes as part of the models' input. We prove that textual descriptions of the items provide useful insights to the systems. 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. Not only these insights are the inspiration for our proposed modifications, but also they suggest possible directions to further improve the models.

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, which means that further boosting the models is going to be even more challenging.

Table of contents

List of figures **xv**

List of tables **xvii**

1 Introduction **1**
1.1 Motivation . . . 2
1.2 Contributions . . . 2
1.3 Outline . . . 3

2 Introduction to Transformer models **5**
2.1 GPT-2 . . . 6
2.2 GPT-3 . . . 7
2.3 BERT . . . 7
2.4 BART . . . 8
2.5 UNITER . . . 9

3 State-of-the-Art Coreference Resolution **11**
3.1 DSTC10 Challenge . . . 11
3.1.1 Overview . . . 11
3.1.2 Problem definition . . . 11
3.1.3 Evaluation metrics . . . 12
3.2 SIMMC2.0 Dataset . . . 12
3.3 Top-performing models of DSTC10 . . . 15
3.3.1 GPT-2 Baseline . . . 15
3.3.2 UNITER-based model for MM Coreference Resolution . . . 15
3.3.3 BART-based model for MM Coreference Resolution . . . 18
3.3.4 DSTC10 competition summary and model replication . . . 23

4 Experiments and Evaluation **27**
4.1 Comparison of the two SIMMC2 domains 27
4.2 Usage of non-visual and visual attributes 28
4.2.1 Additional non-visual information 29
4.2.2 Additional visual information 30
4.3 Models domain generalization 31
4.3.1 Data division for cross-domain evaluation 31
4.3.2 Cross-domain evaluation 32
4.3.3 Cross-scene evaluation 34
4.4 Further improving the models 36
4.4.1 Evaluating the effect of the reference type 37
4.4.2 Addressing under-prediction 38
4.4.3 Model combination 42
4.5 Discussion 43

5 Human-level performance and Error Analysis **45**
5.1 Description of the devtest random subset 45
5.2 Human-level performance 46
5.3 Error analysis 48
5.3.1 Human annotators errors 55
5.3.2 Analysis summary 57

6 Conclusion and Future Work **59**
6.1 Future work 60

References **63**

Appendix A Experimental settings **67**
A.1 Experiments details 67
A.1.1 Models replication 67
A.1.2 Individual evaluation on each SIMMC2 domain 68
A.1.3 Adding non-visual and visual attributes to the input of BART 68
A.1.4 Cross-domain and cross-scene evaluation 68
A.1.5 Evaluating the effect of the reference type 68
A.1.6 Adding the auxiliary task output head 69
A.1.7 Experiments of model combination 69
A.2 Hyperparameter tuning 69

A.2.1 Considered weights for the UNITER-based model 70
A.2.2 Considered weights for the BART-based model 70

List of figures

2.1 Attention-mechanism unit 6
2.2 BERT, GPT and BART noising schemes 9
2.3 UNITER scheme 10

3.1 SIMMC2 two-phase data collection pipeline 13
3.2 GPT-2 based baseline 15
3.3 UNITER-based model overview 16
3.4 Object embeddings of the UNITER-based model 17
3.5 BART-based model overview 22
3.6 Detailed BART diagram 25

4.1 Average number of object per scene in each of the SIMMC2 domains . . . 28
4.2 Object distribution comparison between domains 28
4.3 Detailed BART diagram using non-visual attributes 30
4.4 Average number of objects per scene in each of the cross-domain datasets. . 35
4.5 Object distribution comparison between two cross-domain datasets 36
4.6 Detailed UNITER diagram adding auxiliary head 39
4.7 Detailed BART diagram adding auxiliary head 40

5.1 Dialog examples 46
5.2 Error examples of under-predicting 50
5.3 Error examples of over-predicting 51
5.4 Error examples of relative locations 52
5.5 Error example of not enough focus on conversation context 53
5.6 Error examples of wrongly annotated dialogs 54
5.7 Error example of a bad camera angle 54
5.8 Error example of a short dialog window 55
5.9 Human error because of SIMMC2 wrong annotation 56
5.10 Human error because the difficulty of the scene 56

5.11 Human error because the difficulty of the scene (II) 57

List of tables

3.1 SIMMC 2.0 Dataset Statistics 13
3.2 SIMMC 2.0 Sub-sets Statistics 14
3.3 Replication results summary 26

4.1 Performance comparison on each of the SIMMC2 domains 27
4.2 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 4.3. BART† corresponds to the BART model that uses only the coreference head, previously commented in Section 3.3.4. 31
4.3 Data division of SIMMC2 for cross-domain evaluation 32
4.4 Cross-domain evaluation 33
4.5 Performance of the UNITER and BART-based models in datasets with different scene and object overlapping 35
4.6 Model performance depending on the conversation context 37
4.7 Auxiliary task head results 41
4.8 Combined models results 42

5.1 Statistics of devtest random subset 46
5.2 Performance of human annotators on a random subset of the devtest set . . 47
5.3 Inter-Annotator Agreement by Cohen's Kappa 47
5.4 Performance of investigated models on a random subset of the devtest set . 48

A.1 Loss weight tuning for the UNITER-based model 70
A.2 Loss weight tuning for the BART-based model 70

Chapter 1

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 Soon et al. (2001) tackles coreference resolution of noun phrases in unrestricted text, and Bergsma and Lin (2006) 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 (Kottur et al., 2018; Ramanathan et al., 2014) or visual question answering (Seo et al., 2017).

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.

Throughout the past decades, many approaches have been used to tackle coreference resolution using rich multimodal inputs. Zheng et al. (2018) propose a semi-supervised learning model to correlate embedding space structures of each modality in coreference resolution. Kumar et al. (2017) analyze whether a gesture co-occurs with a specific request or with the context surrounding the request with the goal of detecting gestures that provides extra information for the dialog. Additionally, the 10th Dialog System Technology Challenge considers multimodal coreference resolution as part of one of its tracks for the 2021 edition (Kottur et al., 2021), that will be further described in Sections 3.1 and 3.2. The previous edition, DSTC9, also proposed a track for situated multimodal conversational AI (Moon et al., 2020). Among the huge amount of machine learning models trying to solve coreference resolution, large natural language models, usually known as *Transformers* (Vaswani et al., 2017), are shown to be the best performing systems.

1.1 Motivation

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.

1.2 Contributions

The main contributions of this thesis are as follows:

1. A review and analysis of existing state-of-the-art multimodal coreference resolution systems proposed for the DSTC10 competition in 2021 in Huang et al. (2021) and in Lee et al. (2021).

2. An examination of the current models on challenging scenarios, evaluating their adaptability to the scenes or domains that they were not trained on.

3. A set of improvements for the state-of-the-art models. We enhance the model performance showing an improvement over the winner of DSTC10 challenge.

4. An exhaustive study about the most common errors. We expose the most frequent examples where the models are struggling and we propose possible solutions.

5. Human-level performance estimation. We experiment with the dataset employed in the project and some annotators are asked to perform coreference resolution on a set of randomly picked examples to estimate the human-level performance.

1.3 Outline

This thesis is structured as described below:

In Chapter 2 we introduce the main Transformer models, that are the core of the existing coreference resolution systems.

Chapter 3 presents the DSTC10 competition, the SIMMC2 dataset considered in the project and introduces two state-of-the-art models proposed in the DSTC10 challenge.

The experiments carried out during the development of the project and the obtained results are described in Chapter 4.

In Chapter 5 we estimate the human-level performance and we illustrate common errors of the analyzed models.

To conclude, Chapter 6 summarizes the most important outcomes of the project and it provides some directions for future work.

Chapter 2

Introduction to Transformer models

Many dialog oriented system often used different types of recurrent neural networks (RNNs) due to the sequential nature of the data (Wen and Young, 2020). However, it has been shown that large language models based on self-attention mechanism, commonly known as Transformers (Vaswani et al., 2017), are superior in performance while being more parallelizable allowing training on larger datasets. Attention is transformer's basic component and it is the main differentiable feature of its architecture compared to RNNs. The attention mechanism can be described by Equation (2.1) and allows Transformers to focus simultaneously on different parts of the sequential data, being able to capture dependencies more easily having long-term memory at the same time.

$$Attention(Q,K,V) = \textbf{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2.1}$$

The matrices Q, K and V are called *query*, *key* and *value* respectively, and contain the vector representation of each word in the sequence. In Equation 2.1, we compute how each word in the sequence (represented by Q) is influenced by all the other words in the sequence (represented by K). Additionally, the softmax function is applied to generate a probability distribution between 0 and 1. Then, the result is applied to all the words in the sequence introduced in V. This is repeated several times to allow the system to learn from different representations of Q, K and V. The attention unit described before and in Equation 2.1 is illustrated in Figure 2.1.

Other positive aspect of Transformers is they are trained using *self-supervised learning*. In contrast to supervised learning, where large amounts of data are labeled, in self-supervised learning the data is not annotated and the models infer relationships directly on it. Then, Transformers can be further trained (fine-tuned) in specific tasks using supervised learning.

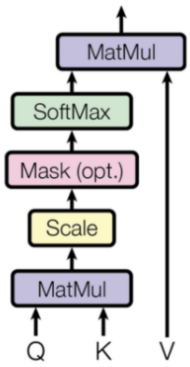


Fig. 2.1 Overview of the attention-mechanism unit that is the core of a Transformer model. Image from Vaswani et al. (2017).

Next, we describe some state-of-the-art pre-trained transformer systems with self-supervision, that can be fine-tuned to address downstream tasks, such as coreference resolution.

2.1 GPT-2

The typical approach to tackle problems using supervised learning is to collect a dataset of training examples that demonstrate reasonable behavior on the task of interest, train a system to imitate that behavior and then evaluate the system performance on a held-out test set identically distributed. However, if the task to solve is formed by the conjugation of complex sub-tasks, this approach is likely to suffer a lack of generalization. This issue is common in NLP tasks, such as machine translation, summarization, question answering or reading comprehension, which are often tackled with supervised learning on task specific datasets.

Nowadays it is thought that training and measuring performance on a wide range of domains and tasks is required in order to model robust NLP systems. One proof of that is the 1.5B parameter Transformer GPT-2 (Radford et al., 2019), trained on 8 million web documents ($\sim$ 40GB of text) that achieves state-of-the-art results of several language modeling tasks in a zero-shot setting.

GPT-2 was trained simple to predict the next word given a sequence of previous tokens (words or sub-words). Some systems incorporate multitasking at an architectural level or at an algorithmic level, but as suggested in McCann et al. (2018), GPT-2 uses natural language to specify the task, inputs and outputs all as a sequence of symbols. For example, a translation training example can be specified as (`translate to Spanish, English text, Spanish text`). Learning from natural language directly seems a large step, nevertheless the internet passively contains a huge amount of dialog information and (Radford et al., 2019) speculates that a large language model with sufficient capacity will be able to learn

to solve and perform many tasks demonstrated in natural language sequences. GPT-2 can be considered as one of the first natural language models to perform unsupervised multitask learning with this zero-shot setting.

2.2 GPT-3

The third version of the Generative Pre-trained Transformer, GPT-3 (Brown et al., 2020), is an autoregressive language model that takes as input and generates text. Autoregressive means that it is reliant on previous values in order to predict current value, so GPT-3 outputs a token given the previous sequence of input tokens. Similar to GPT-2, GPT-3 is also pre-trained on a large set of semi-filtered text documents, formed by Common Crawl (Raffel et al., 2019), consisting of 410 billion byte-pair-encoded tokens and making up the 60% of the training mixture, WebText2 (Kaplan et al., 2020) (22%) that is an extended version of WebText used to pre-train GPT-2 (Radford et al., 2019), Books1 (8%), Books2 (8%) and Wikipedia (3%).

The model is not open source, but it is commercially available through OpenAI API. Brown et al. (2020) mention 8 existing different versions of GPT-3 varying in size. The largest model consists of 175B parameters, 10 times more than the largest version of GPT-2. This difference in size is also translated in performance compared with other language Transformers on different NLP tasks, such as translation, summarization or question answering.

In general, Transformer models can adapt to different tasks by being fine-tuned on a task-specific datasets. However gathering and preprocessing all the necessary data can be tedious. One of the most powerful features of GPT-3 is that it can perform new tasks that it has never been trained on by showing it a few examples of the task. This is called *few-shot learning*, which can be described as the process of feeding a learning model with very little training data. In fact, *one-shot learning* or *zero-shot learning* can be applied to GPT-3 achieving state-of-the-art results in many tasks.

2.3 BERT

The previous Transformer systems are unidirectional models, that is they are predicting the next token as a function of previous (left) tokens. In contrast, Google's BERT (Devlin et al., 2019), which stands for Bidirectional Encoder Representations from Transformers, is a designed to incorporate bidirectional pre-training for language representations. To do so, it uses a *masked language model* (MLM) pre-training objective, i.e., some tokens of the

input text are randomly masked and the goal is to predict the original vocabulary id of the masked word based only on its context. Unlike other systems based on left-to-right language model pre-training (e.g. GPT-2 or GPT-3), the MLM objective allows the representation to be conditioned on the left and the right context at the same time.

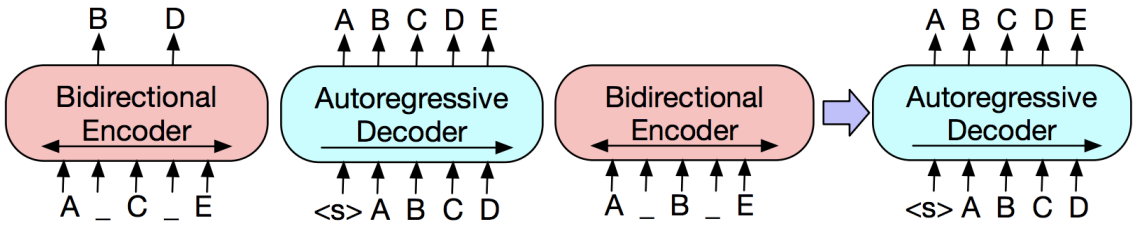
The architecture of BERT is a multi-layer bidirectional Transformer encoder based on the original version of Vaswani et al. (2017). BERT is publicly available and initially it had two possible model sizes: the 110M parameter `BERT-base` model and `BERT-large`, consisting of 340M parameters. These models were pre-trained on BooksCorpus (800M words) (Zhu et al., 2015) and on text passages of English Wikipedia (2500M words).

BERT, like all the other Transformer models, can be specialized in different downstream tasks by fine-tuning it. Compared to pre-training, fine-tuning is relatively cheap and BERT can achieve state-of-the-art results on many natural language tasks just by being fine-tuned for few hours on a GPU. A good example of that can be found in Devlin et al. (2019), where experiments and results on 11 NLP problems are reported.

2.4 BART

In this section, BART (Lewis et al., 2020), a denoising sequence-to-sequence autoencoder is presented. It is trained on masked text produced by a noising function and it learns to reconstruct it, hence the adjective 'denoising'. This Transformer model can be seen as a generalization of BERT and GPT because of its bidirectional encoder and its left-to-right decoder respectively, as seen in Figure 2.2. The encoder and the decoder are connected by cross-attention, that is each decoder layer performs attention over the final hidden state of the encoder output, potentially allowing the whole system to generate an output closely connected to the original input.

The noising process of BERT is also generalized with BART. In Lewis et al. (2020) the arbitrary transformations that are applied include token masking (like BERT in Devlin et al. (2019)), token deletion, sentence permutation or document rotation. The best noising approach experimented with in Lewis et al. (2020) is both randomly shuffling the order of the original sentences and in-filling token masking. Therefore, BART is forced to reason more about the overall sentence length and make longer range transformations to the input.



(a) noising on BERT (b) noising on GPT-n (c) noising on BART

Fig. 2.2 In BERT (Devlin et al., 2019), Subfig. 2.2a, random tokens are replaced with masks and the text is encoded bidirectionally. Corrupted tokens are predicted independently, so BERT struggles at generation tasks. In GPT models (Brown et al., 2020; Radford et al., 2019), Subfig. 2.2b, tokens are predicted conditioned on prior tokens, so they can be used for generation tasks more easily, but they cannot learn bidirectionally dependencies. Finally, BART (Lewis et al., 2020), Subfig. 2.2c, joins both worlds and inputs of the encoder do not need to be aligned with the decoder outputs, allowing arbitrary noise transformations. The input of the encoder (masked document on the left) is processed bidirectionally, and then the likelihood of original document (right) is calculated with an autoregressive decoder. Illustrations from Lewis et al. (2020).

The overall topology of BART is quite related to that used in BERT but it has two main differences. First, each layer of the decoder additionally performs cross-attention over the final state of the encoder; and second, BART does not use an additional feed-forward network before word prediction like BERT does. In total, BART has about 10% more parameters than the equivalent size of BERT. This is also translated in performance: the results of Lewis et al. (2020) show that BART generally achieves the most consistent performance across all investigated tasks.

2.5 UNITER

The Transformers seen so far are natural language models that only accept text-like inputs. However, many tasks require preprocessing and handling visual and language features, such as visual question answering (VQA) or automatic image captioning. LXMERT (Tan and Bansal, 2019) is an example of a Transformer-based model that learns vision-and-language connections. A more modern alternative is UNITER (Chen et al., 2020), that stands for UNiversal Image-TExt Representation, and it is a large-scale pre-trained system that is able to jointly process visual and textual inputs.

The core of UNITER is based on the Transformer model proposed in Vaswani et al. (2017) to use the self-attention mechanism designed for learning contextualized representations.

As illustrated in Figure 2.3, the model encodes visual features (images and bounding box features) and textual features (tokens and positions) into a common high-dimensional space employing a image embedder and a text embedder respectively. Then, a Transformer module is applied to learn generalizable contextualized embeddings.

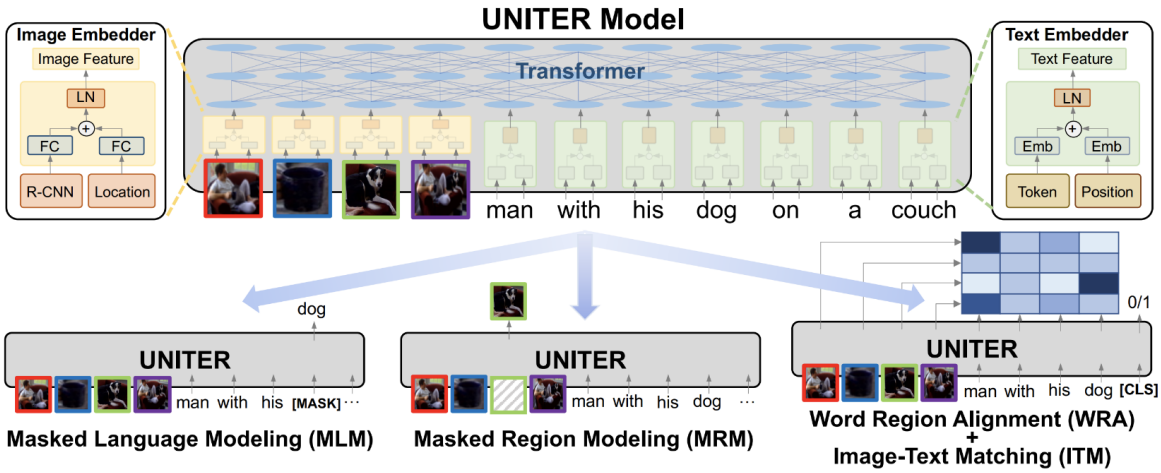


Fig. 2.3 Overview of UNITER, Transformer-based model with multimodal inputs. It consists of an Image Embedder (left), a Text Embedder (right) and a multi-layer Transformer module (center), pre-trained on three different vision+language tasks. Representation from Chen et al. (2020).

UNITER is pre-trained in three different tasks: Masked Language Modelling (MLM) conditioned on image, Masked Region Modelling (MRM) conditioned on text, and joint Image-Text Matching (ITM). Chen et al. (2020) has proven that conditioning the tasks with respect the other modality is beneficial for the overall system performance since this strategy can ease the missing-alignment between images and text and obtain better embedding representations for downstream tasks.

To evaluate the performance of UNITER, it has been tested on six vision-language tasks across nice datasets, including Visual Question Answering, Visual Commonsense Reasoning, Natural Language for Visual Reasoning, Visual Entailment, Image-Text Retrieval and Referring Expression Comprehension. Experiments show that UNITER achieves new state-of-the-art results in all six downstream tasks (Chen et al., 2020).

Chapter 3

State-of-the-Art Coreference Resolution

This chapter is focused on characterizing the state-of-the-art coreference resolution. The DSTC10 challenge is introduced in Section 3.1 and its involvement in coreference resolution through SIMMC2 track in Section 3.2. The best performing models on coreference resolution of the DSTC10 competition are described and replicated in Section 3.3.

3.1 DSTC10 Challenge

3.1.1 Overview

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 3.2).

3.1.2 Problem definition

A multimodal dialogue system that uses language and visual information in communication with an user should be able to identify the objects in the visual scene that the user refers to in the dialog. Assuming there is a bounded number of candidate objects, i.e., all the objects present in the scene, we can formulate the coreference resolution task as a *binary classification problem* on each possible item.

3.1.3 Evaluation metrics

The ultimate goal of this project is to solve the coreference resolution task with the highest possible performance. Here and in the DSTC10 competition, performance is evaluated using *micro F1-score*, that is the element-wise harmonic mean of precision and recall, as computed by Equation (3.3). Note that one utterance can have various referred items, and the *micro* F1 score evaluates the performance on each referred object individually.

The *precision* (eq. 3.1) is the ratio of true positives among all predicted positive examples, and *recall* (eq. 3.2) is the ratio of true positives among all ground-truth positive examples.

$$precision = \frac{true\ positives}{predicted\ positives} = \frac{true\ positives}{true\ positives + false\ positives}. \tag{3.1}$$

$$recall = \frac{true\ positives}{ground\text{-}truth\ positives} = \frac{true\ positives}{true\ positives + false\ negatives}. \tag{3.2}$$

$$\textbf{F1 score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.3}$$

3.2 SIMMC2.0 Dataset

The second track of the DSTC10 challenge considers the tasks proposed in Kottur et al. (2021) 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 (Moon et al., 2020), 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.

The dataset 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

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

per scene and they might be visually overlapped when hanging on walls or on shelves. The Table 3.1 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 3.1 SIMMC 2.0 Dataset Statistics

The dialogs of SIMMC2.0 dataset were generated through a two-stage data collection pipeline (Fig. 3.1). 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.

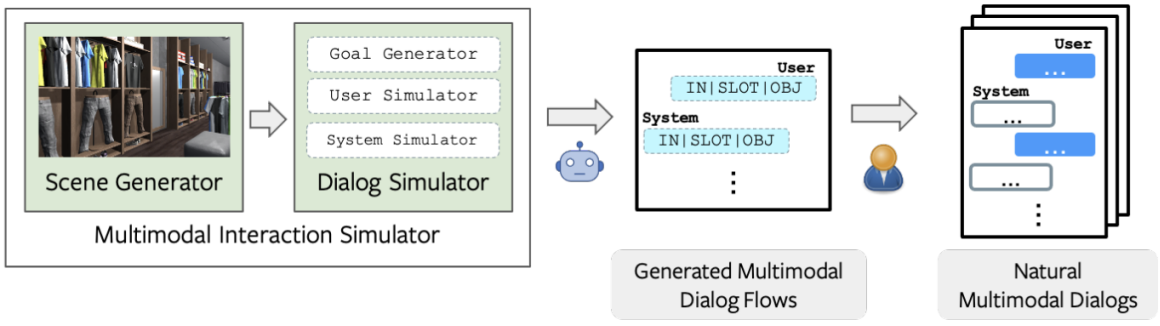


Fig. 3.1 SIMMC2 two-phase data collection pipeline. In phase 1 a situated multimodal dialog simulator generates the visual environment and the dialog. In phase 2 the dialogs flows are paraphrased by human annotators accordingly a real word application. Representation from Kottur et al. (2021).

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 3.2. 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 modelling.
Dev	5%	563	Used for hyperparameter selection and other modelling 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 3.2 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. The research team also proposes a GPT-2 based system that tackles all these tasks and it will be explained in Section 3.3.1. The performance of this system in multimodal coreference resolution will be considered as a baseline for this project.

3.3 Top-performing models of DSTC10

3.3.1 GPT-2 Baseline

The baseline considered in this project is a GPT-2 based system proposed in Kottur et al. (2021). Originally, the system was modeled to study Dialog State Tracking (DST) task using SIMMC1 dataset (Moon et al., 2020). However, the system is enhanced using the joint supervision signals for the MM-Disamb, MM-Coref, DST and Response Generation tasks.

As illustrated in Figure 3.2, the GPT-2 based model takes as input the dialog history and the flattened multimodal context to predict the belief states and other responses, such as the referred objects in the last user turn, that is the output of interest in this piece of work.

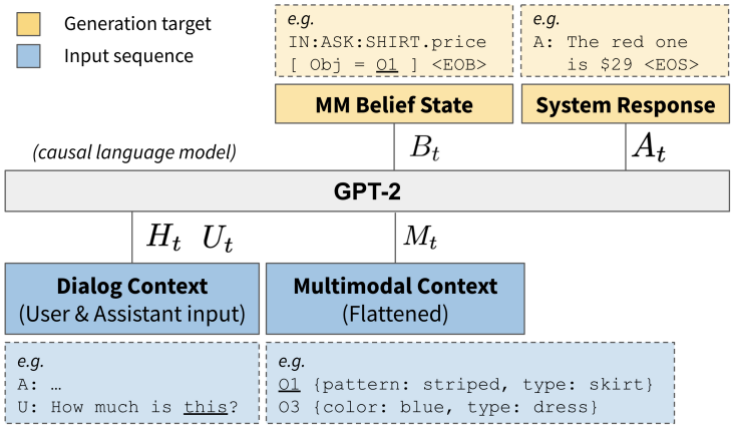


Fig. 3.2 Illustration of GPT-2 based baseline for MM-Coref task (among others). It takes as input the dialog history as well as the flattened multimodal context. The referred objects in the last user turn are included in the output. Illustration from Kottur et al. (2021)

The performance on MM-Coref of this baseline is **0.366 F1-score** (Table 3.3). This is expected to be a baseline easy to beat, since the employed version of GPT-2 is the 12-layer pre-trained language model of 117M parameters, way smaller than the 1.5B parameter GPT-2 introduced in Section 2.1.

3.3.2 UNITER-based model for MM Coreference Resolution

This section presents the UNITER-based system (Huang et al., 2021) to tackle multimodal coreference resolution that ranked second in this task in the DSTC10 competition.

As summarized in Section 2.5, UNITER (Chen et al., 2020) is a Transformer-based model for image and text universal embeddings, that is pre-trained in three different tasks on

visual+language data: masked language modeling, masked region modeling and word-region alignment. In Huang et al. (2021), UNITER is extended to handle the rich multimodal inputs of SIMMC2.0 dataset (Kottur et al., 2021).

Model overview

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 proposed model in Huang et al. (2021) 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. Figure 3.3 illustrates the overview of this system.

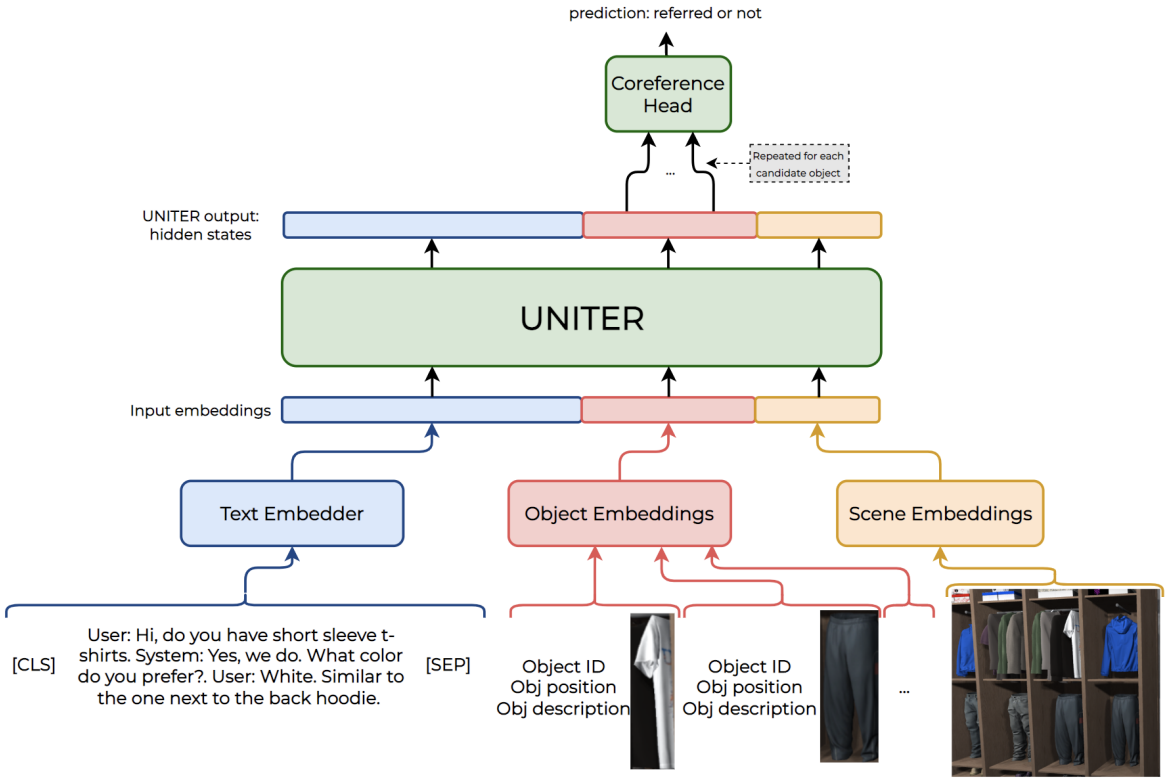


Fig. 3.3 Overview of the proposed UNITER-based system.

The dialog history is preprocessed using the BERT (Devlin et al., 2019) tokenizer, that translates text to language tokens. Object and scene embeddings are modeled as described below.

Object embeddings

Each object is encoded using several multimodal features as shown in Figure 3.4. The features are preprocessed as follows:

- The object scene-level index is embedded through an embedding layer.
- The scene image is first cropped to contain just the object and it is fed into a visual encoder to get the visual embedding. In Huang et al. (2021) two visual encoders are investigated: the pre-trained CLIP (Radford et al., 2021) model and BUTD (Anderson et al., 2018).
- The 3D position of the object (3-dimensional coordinates) of the object are used without any preprocessing.
- The non-visual object metadata, also known as non-visual knowledge base (KB) entry, is encoded using a text encoder to get its embedding representation. Again, in Huang et al. (2021) two text embedders are tested: BERT (Devlin et al., 2019) and sentence BERT (SBERT) (Reimers and Gurevych, 2019).
- A binary feature, `scene_active`, is employed to indicate whether an object is in the current active scene. Similar to the object index, this label is encoded through a embedding layer.
- Another binary feature, `prev_mentioned`, is incorporated to indicate whether an object has been mentioned in previous dialog turns. The label is encoded through a embedding layer.

The final object embedding is produced after feeding all these features into a dense layer. Note that all the non pre-trained embeddings are trained in an end-to-end manner.

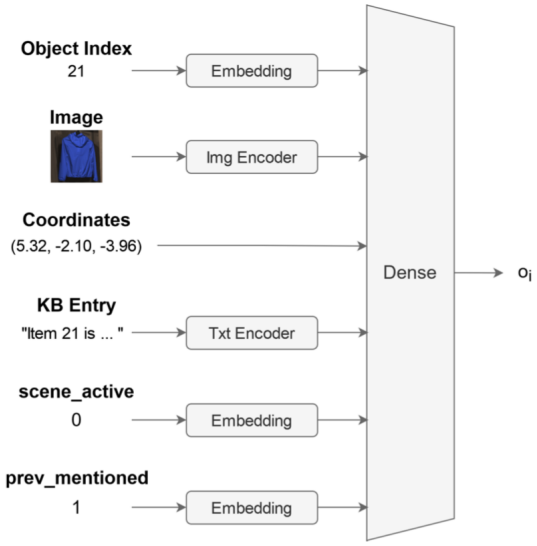


Fig. 3.4 Object embeddings of the UNITER-based model. Scheme from Huang et al. (2021).

Scene embeddings

Embeddings of the whole scene images are also added to reflect visual information that is not included in the cropped object images. 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.

Multimodal UNITER encoder

After computing dialog history U encoding, object embeddings O and scene embeddings S, they are all fed into an implementation of UNITER[2]. 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 3.4.

$$H = \text{Encoder}(U, O, S) \longrightarrow Z = \text{Dense}(H) \longrightarrow Y = \sigma(Z). \tag{3.4}$$

Model reported performance

Several combinations of features for the object embeddings (Fig. 3.4) and scene embeddings were investigated for the DSTC10 competition. The described model here is reported to achieve 0.728 F1 score on the `devtest` set. However, this is not the configuration submitted by Huang et al. (2021) for the DSTC10 competition. The best performing model was an ensemble-based system comprising 5 individual UNITER-based models with different feature configurations, resulting in 0.741 F1 score. The best individual system of this ensemble achieved **0.674 F1 score**. These two models differ just in the fact that the later is not using the `prev_mentioned` feature that indicates the object was previously mentioned in the dialog. This results and its equivalent replication can be found at Table 3.3.

3.3.3 BART-based model for MM Coreference Resolution

The traditional approach for developing task-oriented dialog systems consists of a pipeline that several modules work together to integrate natural language understanding (NLU), dialog state tracking (DST), dialog policy management (DPM) and natural language generation. In

[2]The implementation of UNITER used in Huang et al. (2021) to build the final system can be found in https://github.com/YIKUAN8/Transformers-VQA and it is not the proposed implementation by the original UNITER paper (Chen et al., 2020).

this section it is reviewed the system proposed in Lee et al. (2021) that participated in the DSTC10 challenge.

Model overview

The system is based on BART Transformer model (Lewis et al., 2020) and several task-specific heads are attached to its outputs so that the model can perform all subtasks at the same time. The 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). The reasoning behind this is that, first, the vision models are usually pre-trained on natural images and fine-tune them would require a large number training examples of 3D images. Second, in a realistic scenario where the virtual assistant is deployed, the object metadata and scene graphs would be readily available and training an additional vision model would be an unnecessary overhead. Additionally, extra supervision signals at training time can still be incorporated for modality alignment using the available metadata.

Defining how the SIMMC2.0 data is encoded to be fed as input for the BART-based system is an important task and Lee et al. (2021) use the following features.

Input representation

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. SIMMC2.0 assumes that utterances may mention objects that are not in the current scene S_T but in the previously observed scene $S_{t<T} \neq S_T$. Hence, the model integrates the objects from the previous scene that are not in the current scene.

Canonical object ID token

A canonical object ID token takes the form of <\d+> (e.g. <32>). This is a scene-level ID and it provides a relational context of the object within the scene, grounding each object to its

scene object index provided in the dataset. This feature to encode an object is also used in the GPT-2 baseline (Section 3.3.1), but the BART-based system also integrates this token along with global IDs to provide contextual information of the object and its absolute attributes.

Unique object ID token

A unique object ID token takes the form of `<{domain}_\d+>` (e.g. `fashion_123`). The digits following the domain specifier denote the index of the unique object in that domain. This token intends to capture visual and non-visual attributes of each object.

Separator tokens

Separator tokens are often used to delimit different parts of the input for several Transformer models. They are special tokens that the model should distinguish from natural words in the input text. In this case, `<SOM>` and `<EOM>` are used to delimit the start and end of the multimodal context; `<SOO>` and `<EOO>` for the start and end of scene objects; `<OBJ>` is employed to differentiate each object in the scene, which are represented by the concatenation of its canonical object ID token and its unique object ID token. Also, the token `<PREVIOBJ>` masks the objects from the previous scene instead of `<OBJ>`. Finally, `<SOB>` and `<EOB>` tell where the belief state starts and ends.

Object locations

The locations of each object are also provided in the input to allow the model infer spatial relations among objects within the scene. To do so, object locations are encoded using common used techniques in VL models (Li et al., 2020; Zhang et al., 2021). 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.

Learning process

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 $S_{t \leq T}$. 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. A simple cross-entropy loss is considered for MM-Coref, denoted as $\mathscr{L}_{\text{mm-coref}}$.

As said before, the BART-based model tackles all SIMMC2.0 subtasks at the same time, so it formulates similar losses for the other tasks. Since this project is not focused on those they are omitted, but the total loss of the model is described by Equation (3.5), where $\mathscr{L}_{\text{LM}}$ is the standard left-to-right LM loss (Bengio et al., 2003) used to approach MM-DST and response generation subtasks.

$$\mathscr{L}_{\text{task}} = \lambda_{\text{LM}}\mathscr{L}_{\text{LM}} + \lambda_{\text{mm-disamb}}\mathscr{L}_{\text{mm-disamb}} + \lambda_{\text{mm-coref}}\mathscr{L}_{\text{mm-coref}} + \lambda_{\text{retrieval}}\mathscr{L}_{\text{retrieval}} \quad (3.5)$$

Auxiliary heads

Additional heads are considered to provide additional signal for coreference resolution. The first extra task, Empty-Coref, aims to predict whether the current dialog turn has MM-Coref targets. The MM-Coref head might predict that there are targets when there is actually none, so whenever the Empty-Coref head prediction is true, i.e. there is no coreference target, the predicted referred objects are overwritten to false. An extra special token, `<NOCOREF>`, has to be added for pooling. For training it is considered the binary cross-entropy loss $\mathscr{L}_{\text{empty-coref}}$.

Object attributes are also encoded by training to classify each object to its corresponding visual and non-visual attributes. Each additional attribute head predicts a categorical class for each corresponding object. Let $\mathscr{O}_{t\leq T}$ be the set of objects in the scene history $S_{t\leq T}$. The attribute multi-class classfication loss $\mathscr{L}_{\text{att}}$ for all objects in $\mathbb{O}_{t\leq T}$ is:

$$\mathscr{L}_{\text{att}} = \sum_{j\in\mathbb{O}_{t\leq T}} \sum_{k=1}^{K} \sum_{c\in C_k} -\mathbb{1}\{c = y_{jk}\} \log P(c),$$

where K is the number of attributes, C_k the set of all classes for the k-th attribute and y_{jk} the label of the k-th attribute of the j-th object. The total loss of the model after adding heads to provide additional signal for coreference resolution is described by Equation (3.6).

$$\begin{aligned}\mathscr{L}_{\text{total}} = \lambda_{\text{LM}}\mathscr{L}_{\text{LM}} + \lambda_{\text{mm-disamb}}\mathscr{L}_{\text{mm-disamb}} + \lambda_{\text{mm-coref}}\mathscr{L}_{\text{mm-coref}} + \\ + \lambda_{\text{retrieval}}\mathscr{L}_{\text{retrieval}} + \lambda_{\text{att}}\mathscr{L}_{\text{att}} + \lambda_{\text{empty-coref}}\mathscr{L}_{\text{empty-coref}}\end{aligned} \quad (3.6)$$

The illustration 3.5 represents the whole proposed model in Lee et al. (2021) and summarizes visually the previous description in this section.

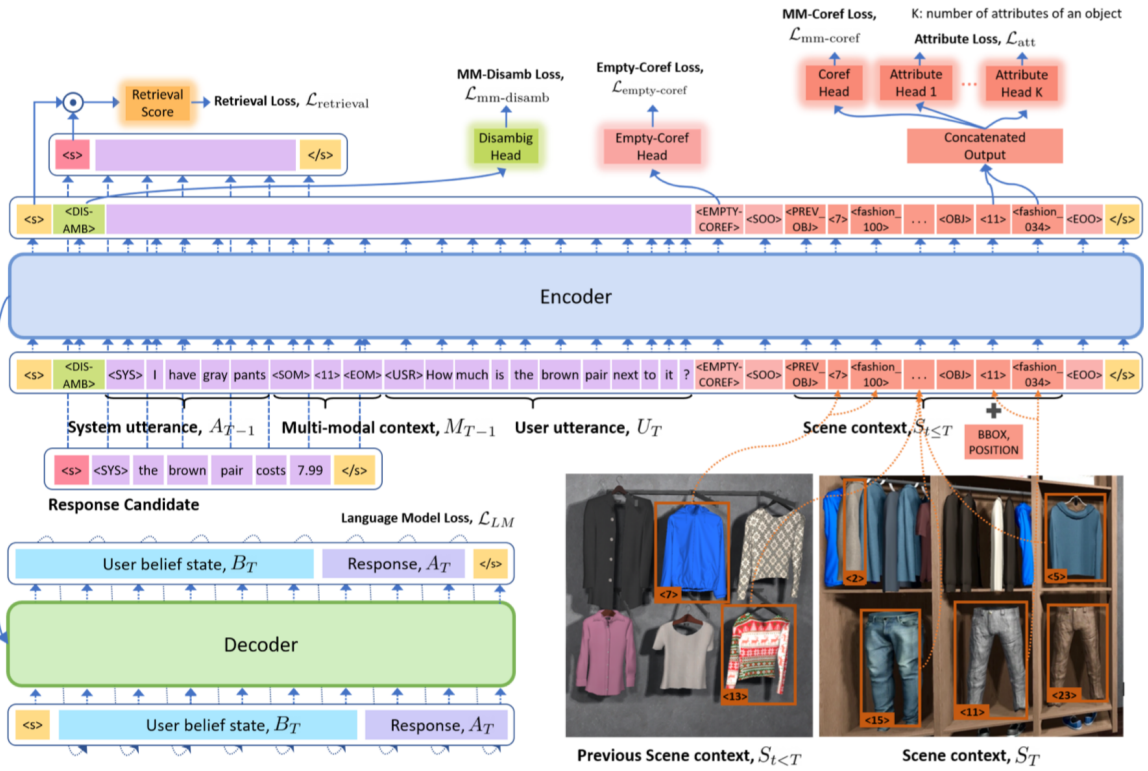


Fig. 3.5 Overview of the BART-based model (Lee et al., 2021) that is jointly trained on four tasks. In this project we are just focused on the *Coref Head*.

Model reported performance

The results on the `devtest` (validation) set of the BART-based model were extraordinary. The proposed system was the winner in MM-Coref and response retrieval subtasks, and the runner-up solution for MM-Disamb, MM-DST and response generation subtasks. This project focuses on multimodal coreference resolution, and the result on this subtask was **0.743 F1 score**, being the first ranked team in the DSTC10 competition on it. In a future section the model will be replicated, illustrated in Table 3.3.

Ablation studies

Ablation studies were conducted on auxiliary task heads for MM-Coref. Deleting attribute classification heads resulted in a considerable drop in the MM-Coref performance by 6.0%. Removing just Empty-Coref target showed no significant difference from the full model. However, Empty-Coref prediction seems to be crucial when attribute classification targets are also suppressed, dropping an additional $\sim$ 20% in performance.

3.3.4 DSTC10 competition summary and model replication

GPT-2 Baseline replication

The GPT-2 Baseline introduced in Section 3.3.1 can be easily replicated since the code is included in the SIMMC2.0 GitHub repository[3].

The model can be trained using the default parameters on a GPU, and then evaluated using the provided evaluation script. The resulting performance is **0.381 object F1 score** for multimodal coreference resolution, quite similar to the 0.366 F1 score reported in Kottur et al. (2021). This system was the first baseline for the DSTC10 challenge, but it is not considered or further studied in this project, since the two following approaches outperform it.

UNITER-based model replication

The UNITER-based model proposed in Huang et al. (2021) is replicated using the open source code provided by the authors[4]. The repository also contains the preprocessed dataset used by this team. This is quite helpful since the dialog, objects and images embeddings are pre-computed, saving time and computing resources.

The best individual system submitted to the DSTC10 competition that achieved **0.675 F1 score** can be replicated using the default parameters: the overall system is trained using focal loss (Lin et al., 2017) with $\gamma = 2$ and $\alpha = 1$ for the negative class (in this case, when an object is not referred), and $\alpha = 5$ for the positive class (i.e., the object is referred). It is used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $5 \cdot 10^{-6}$ and $\varepsilon = 10^{-8}$, and a batch size of 16. The team trained the model for a maximum of 30 epochs and performed early-stopping as a function of the F1 score on the SIMMC2.0 dev set.

In addition, this model can be further enhanced by considering previously mentioned objects in the dialog. This is also replicated, getting a final **0.726 F1 score** on the `devtest` set. Finally, the initial model in Huang et al. (2021) was not using object IDs and after including them the performance slightly dropped. Moreover, 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. Replicating the model removing the usage of object IDs and considering previously mentioned objects in the dialog, the resulting performance is **0.758 F1 score**. All the future experiments in Chapter 4 will prioritize the latest for investigation.

[3]Repository located at https://github.com/facebookresearch/simmc2
[4]Code publicly available at https://github.com/i-need-sleep/mmcoref_cleaned

BART-based model replication

The BART-based model (Lee et al., 2021) ranked first in the DSTC10 competition on the coreference resolution task and can be replicated using the publicly available code[5]. The repository repository contains the preprocessed dataset and a link to download the trained model reported in Lee et al. (2021). In this section, we describe the model's learning process. The detailed diagram 3.6 describes each stage of the BART-based system. For every natural language example, formed by the concatenation of special tokens (delimiters), dialog history, object IDs and object encoded locations, the BART-based model first extracts the object locations (marked with blue color) from the rest of the example. The encoded locations are embedded through a single-layer fully-connected neural network of input size 6 (dimension of each encoded location) and output size 1024, that is the input dimension of the BART encoder. The rest of the example is tokenized and then passed through the BART encoder embedder to obtain the 1024-dimensional representation of each token. The BART encoder embedder ("BART encoder" in the illustration, for the sake of brevity) is trained separately from the rest of the system to represent the global object IDs close to their corresponding object attributes in the high-dimensional embedding space. This is computed similar to the image-caption encoding technique of Radford et al. (2021): a multimodal embedding space is learned by maximizing the cosine similarity between object global IDs embeddings and their corresponding attributes, and also minimizing the cosine similarity between the global IDs embeddings and the attributes that do not correspond to the object. Given the tokenized object IDs, they are embedded through the trained BART embedder and the resulting embeddings are summed with the embeddings of the object locations before feeding the BART encoder. The output of BART is the encoding of each embedded token. The encodings corresponding to each object canonical ID and global ID (each of them of size 1024, so the concatenation is a 2048-dimensional vector) are finally used as input in the coreference head, that is a single-layer fully-connected neural network of input size 2048 and output size 2. The output of this head determines if the object was referred or not.

After setting up the codebase, we can train and evaluate the whole model again to fully replicate it using the default parameters. The core of the model is a 24-layer BART Transformer model. It is fine-tuned for 10 epochs with an initial learning rate of $5 \cdot 10^{-5}$ and a batch size of 16 with AdamW optimizer (Loshchilov and Hutter, 2018). The best checkpoint evaluated at every 1000 steps is chosen on the `devtest` set. The final choice of joint learning coefficients are:

$\lambda_{\text{LM}} = 1.0$; $\lambda_{\text{mm-coref}} = 0.8$; $\lambda_{\text{retrieval}} = 0.4$; $\lambda_{\text{mm-disamb}} = \lambda_{\text{att}} = \lambda_{\text{empty-coref}} = 0.1$

[5] Available in GitHub at https://github.com/KAIST-AILab/DSTC10-SIMMC/

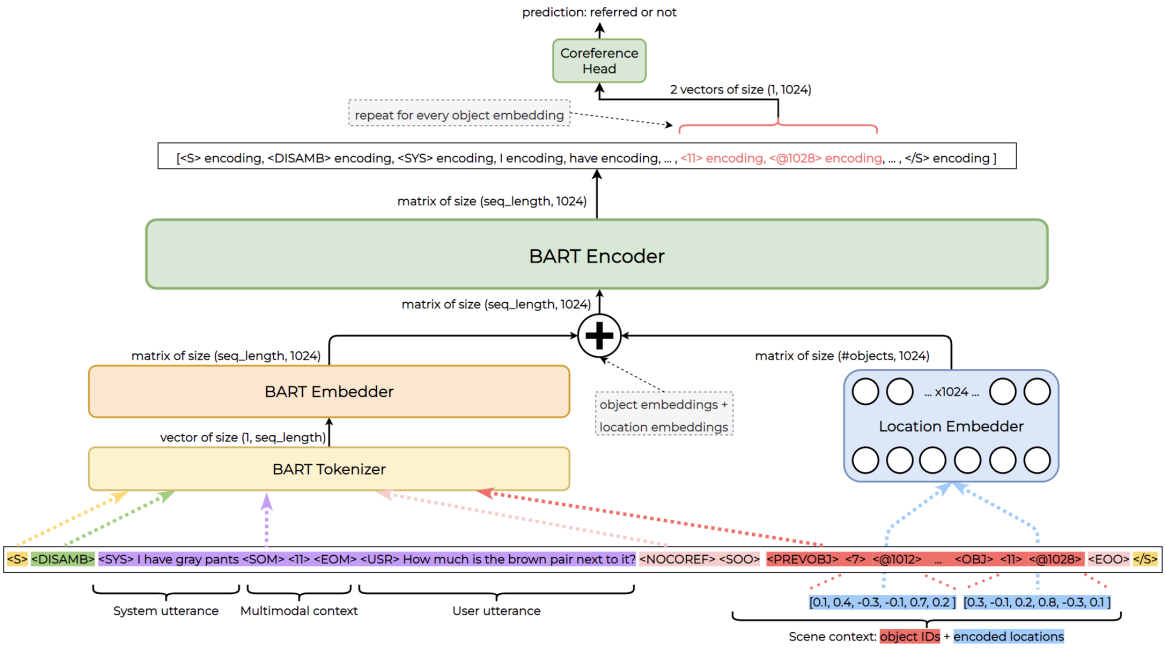


Fig. 3.6 Detailed diagram of the BART-based model proposed in Lee et al. (2021). Before feeding the BART encoder, every natural language example is converted to its embedding representation. The last step is to pass each object encoded embedding through the coreference head, which determines if the object was referred or not.

The resulting performance on the `devtest` set is **0.742 F1 score**, really similar to the value achieved in the original paper, 0.743 F1 score. Since this project is just focused on the multimodal coreference resolution task, suppressing the rest of the task heads is also investigated and a detailed version of the model just keeping the coreference head is shown in Figure 3.6.

A single-head model is also trained and evaluated, increasing the model performance up to **0.748 F1 score**. This means that the BART-based model was not benefiting from solving all the tasks at once as it was hypothesized by Lee et al. (2021). Then, the experiments of this project (Chapter 4) are focused on the single-head version of the BART-based model, so the main concern will be improving on the coreference resolution task.

All the results provided in this section, both the reported by the original paper (Lee et al., 2021) and the ones achieved after the replication, can be found at Table 3.3.

Results summary

The Table 3.3 summarizes the replication results mentioned so far as well as the reported results in the DSTC10 competition.

Model	Description	Original paper	Our replication
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 3.3 Replication results summary. Comparison between reported performance in the original papers (Huang et al., 2021; Kottur et al., 2021; Lee et al., 2021) and our results.

Chapter 4

Experiments and Evaluation

In this chapter the methods explained and replicated in Chapter 3 are investigated. In Section 4.1, we analyze the performance on the two SIMMC2 domains. Next, we propose some modifications to the BART-based model in Section 4.2. We also study how the systems would perform on unknown scenarios in Section 4.3. Finally, in Section 4.4, we further improve the models by including an additional task head and by combining the strengths of the models.

4.1 Comparison of the two SIMMC2 domains

In Section 3.2 we introduced SIMMC2.0 dataset. The 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 analyze the performance of the UNITER-based and BART-based models on each of the domains and the results are available in Table 4.1.

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

Table 4.1 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 reason why this is happening is because the fashion domain is more challenging since the number of object per scene tends to be larger and the items might be visually overlapped. Plot 4.1 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.

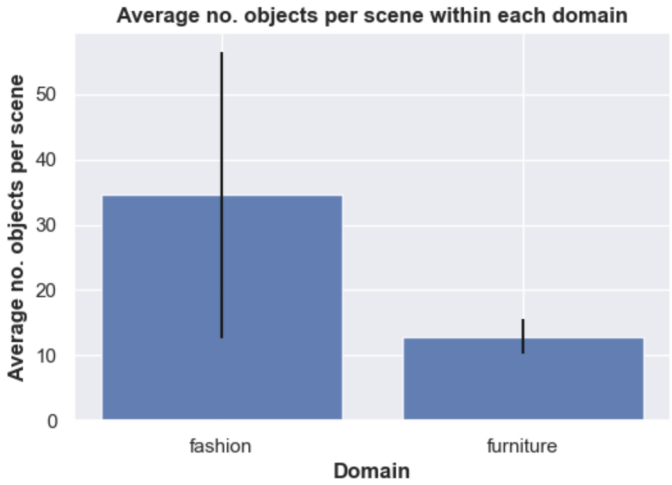


Fig. 4.1 Average number of object per scene in each of the SIMMC2 domains.

The examples of the scenes for the two domains in Figure 4.2 illustrate the difference between them. Fashion stores are often more crowded than furniture shops.

Fig. 4.2 Comparing the object distribution between fashion and furniture domains. Fashion stores (example on the left) tend to be more populated compared to the ones within furniture domain (example on the right).

4.2 Usage of non-visual and visual attributes

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. In this section, we investigate the effect of using these attributes as they were object descriptions. We also consider experiments 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.

Both BART and UNITER-based models use visual and non-visual information. The UNITER-based model considers non-visual attributes as part of the input to generate the object embeddings (Fig. 3.4), and the visual features are extracted from scene images. In contrast, the BART-based model trains the embedder using both non-visual and visual attributes, but it is not considering these attributes as part of the input. Adding non-visual attributes was beneficial for the UNITER-based model (Huang et al., 2021), therefore we explore adding non-visual and visual attributes to the textual input of the BART-based model alongside the object locations and object IDs.

In the following Subsections 4.2.1 and 4.2.2 we investigate the effect of including object descriptions (attributes) as part of the input in the BART-based model.

4.2.1 Additional non-visual information

The BART-based model incorporates both visual and non-visual attributes to train the BART encoder embedder. Lee et al. (2021) applies the technique proposed in Radford et al. (2021) which intends to encode the global object IDs close to each 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 attributes as part of the input like the object location encodings can provide additional signal to link an object to its textual description.

The usage of just non-visual attributes is first investigated, which is legal under the DSTC10 competition rules. Figure 4.3 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. The rest of the process is unaltered as illustrated in Figure 3.6.

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 4.2.

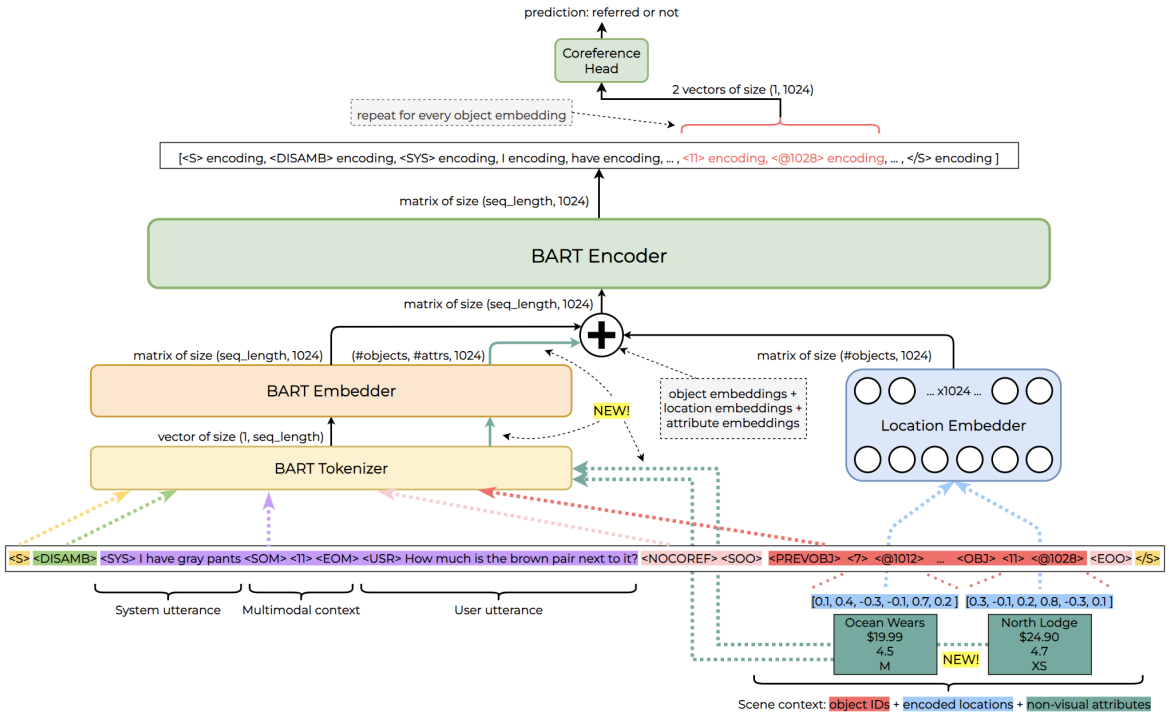


Fig. 4.3 Detailed diagram of the BART-based model proposed in Lee et al. (2021) including additionally non-visual attributes as part of the input along with object locations.

4.2.2 Additional visual information

Although visual attributes from the `metadata` files are banned in the DSTC10 challenge, this project analyzes the benefits of incorporating that piece of information. Similar to the previous model where just non-visual were used (Fig. 4.3), visual attributes are included 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.

Again, 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 4.2.

To sum up, all the results of this section are found in Table 4.2. 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.

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

Table 4.2 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 4.3. BART† corresponds to the BART model that uses only the coreference head, previously commented in Section 3.3.4.

4.3 Models domain generalization

In this section we assess the adaptability of the models to different scenarios. The dialogs of SIMMC2.0 dataset (Kottur et al., 2021) take place in clothing and furniture stores. 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 first study the models behavior on unseen objects (out-of-domain) in Section 4.3.2, and then in Section 4.3.3 we compare the performance across domains, considering scenarios with seen objects but unseen scenes (different physical stores).

4.3.1 Data division for cross-domain evaluation

The experiments of this section require to re-arrange the dialogs of the SIMMC2 dataset, so we can evaluate the models in different domains or in unseen scenes. Table 4.3 describes all the subsets considered for the cross-domain investigation. Note that these sets are not disjoint, the same dialog might be present in several subsets, but we make sure we use a disjoint division for each of the experiments.

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: **IN-DOMAIN** (used for testing) contains dialogs with the same distribution as **FASH-14K**. The **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, **OUT-OF-DOMAIN** uniquely contains dialogs from the furniture domain.

Name	No. examples	Description
FASH-14K	∼ 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	∼ 12K	Set used for training. All SIMMC2.0 furniture domain examples.
FASH-6K	∼ 6.5K	Set of fashion domain examples of the SIMMC2.0 `devtest` set.
FURN-2K	∼ 2K	Set of furniture domain examples of the SIMMC2.0 `devtest` set.
IN-DOMAIN	∼ 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.
IN-DOMAIN HELD-OUT	∼ 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.
OUT-OF-DOMAIN	∼ 9.5K	Used for testing. All examples are within furniture domain. It evaluates the adaptability of the models to different domains.

Table 4.3 Division of the SIMMC2 dataset for cross-domain evaluation.

4.3.2 Cross-domain evaluation

In this section, we evaluate the performance of the models on unseen domains. A multimodal conversational system, deployed on unseen domains or on the same domain with unseen objects, can rely on generic visual (extracted from the image) and non-visual (extracted from meta-information) attributes. Global object IDs in such cross-domain scenarios should not be helpful when testing the model on unseen domains or in the same domain with unseen objects. In Section 3.3.4, we have proven that the UNITER-based model does not need object IDs to perform well. However, global IDs are crucial in the learning algorithm of the BART-based model, since this system is employing global IDs to capture non-visual and visual attributes of the objects using a learning technique similar to the learning algorithm of CLIP (Radford et al., 2021). This technique also enables zero-shot transferability for downstream tasks, so the BART-based model will not break even though the objects at inference time were not seen during training.

Table 4.4 shows the results of the models trained on out-of-domain datasets (out-of-domain) 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 4th and 5th columns that report the result on each test set.

Base model	Attributes	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 4.4 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.

It is also studied the effect of including object non visual attributes as part of the input (NV), that would be valid in the DSTC10 competition, and also both non-visual and visual attributes (NV+V).

Mainly, the UNITER-based model performs better than the BART-based on out-of-domain scenarios. We hypothesize 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. Focusing on the last column, 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 out-of-domain set. The drop in performance of the UNITER-based models is around 30%, whether the BART-based models decrease more than 40% F1 score since they heavily rely on object encodings modeled at training time. It is important to note that 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 out-of-domain. 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 a out-of-domain 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 (more than 50%). Visual attributes help, but this time they are not enough to reach UNITER out-of-domain performance. Definitely, BART-based models cannot be used in out-of-domain scenarios. The UNITER-based models are superior in this aspect, but they are not performing extraordinary well either.

Note that models tend to perform better in the furniture domain. This is because the furniture domain contains less crowded scenes and it can be checked in plot 4.1 and in illustration 4.2. In the furniture domain, the stores tend to contain fewer objects and they are not typically overlapped like in the fashion domain.

4.3.3 Cross-scene evaluation

We continue the study on out-of-domain scenarios, but we also investigate how the models perform on unknown stores (not seen at training time) and different object distributions. To do so, we employ **FASH-14K** set for training, and **IN-DOMAIN**, **IN-DOMAIN HELD-OUT** and **OUT-OF-DOMAIN** sets for testing, all of them described in Table 4.3. Note the distinction between conversations taking place in `cloth_store_1498649_woman` shop and the rest. The objects in a certain store can be placed or distributed differently than in other buildings even though they are all within the same domain (fashion in this case). The **IN-DOMAIN HELD-OUT** set will assess the model performance in seen domains but with different object distributions.

Both UNITER and BART-based models are trained on **FASH-14K** dataset and tested on three different cross-domain sets. It is also studied the effect of including object non visual attributes as part of the input (NV), that would be valid in the DSTC10 competition, and also both non-visual and visual attributes (NV+V). Textual visual attributes were not allowed in the DSTC10 challenge, but in this project it is also investigated what would happen if we had a system that would recognize these visual attributes before performing coreference resolution. The results of these experiments are illustrated in Table 4.5. The BART-based models perform better on **IN-DOMAIN** and on **IN-DOMAIN HELD-OUT** than the UNITER-based model. In fact, BART-based models are performing considerably well when the possible objects were seen at training time. Comparing the performance between **IN-DOMAIN** and on **IN-DOMAIN HELD-OUT** 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 that 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 (**OUT-OF-DOMAIN** set), the UNITER-based models are performing about 8% better since they have introduced some generalization by not relying on object IDs.

Base model	**Attributes**	**IN-DOMAIN**	**IN-DOMAIN HELD-OUT**	**OUT-OF-DOMAIN**
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 4.5 Performance of the UNITER and BART-based models in datasets with different scene and object distributions. The models are trained on **TRAIN-CD** dataset and evaluated in **IN-DOMAIN** (scene and object overlapping), **IN-DOMAIN HELD-OUT** (object overlapping) and **OUT-OF-DOMAIN** (unseen objects). BART† is the BART model that uses only the coreference head, and UNITER† is the improved version after DSTC10 evaluation using previously mentioned objects and no object IDs. The systems may incorporate in the input object non-visual attributes (NV) or both non-visual and visual attributes (NV+V).

One aspect that was not commented yet but can seem odd is the fact that BART-based models are performing better on **IN-DOMAIN HELD-OUT** set than on **IN-DOMAIN** set. This is because the typical settings of the scenes in **IN-DOMAIN HELD-OUT** dataset are easier to analyze, since they have fewer objects as shown in Plot 4.4.

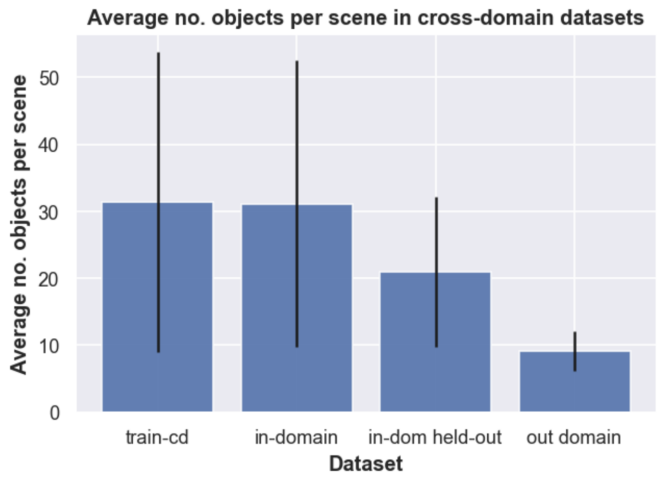
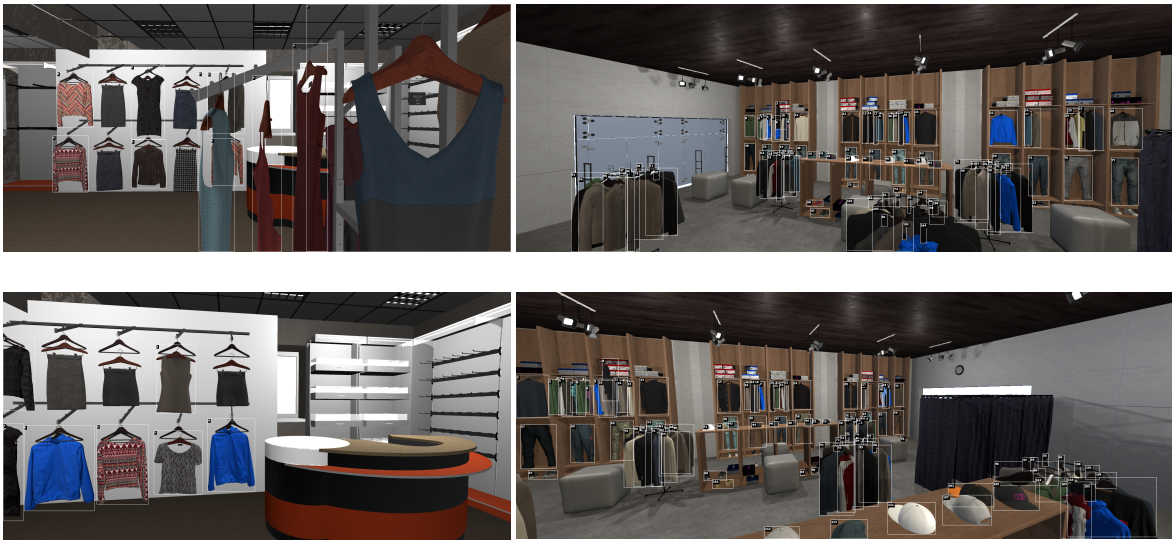


Fig. 4.4 Average number of objects per scene in each of the cross-domain datasets.

The **TRAIN-CD** and the **IN-DOMAIN** sets have on average 30 items per scene, but the **IN-DOMAIN HELD-OUT** set has around 20 objects per scene on average. This is corroborated and exemplified by Figure 4.5, where we expose some scenes of the stores of both sets.



(a) `cloth_store_1498649_woman` store (b) `cloth_store_paul` store

Fig. 4.5 Comparing the object distribution between **IN-DOMAIN** and **IN-DOMAIN HELD-OUT** datasets. The typical shops where the dialogs of the **IN-DOMAIN** dataset take place are much more populated of clothing items. The BART-based model is performing better on the **IN-DOMAIN HELD-OUT** set, where the number of possible referred objects is smaller.

4.4 Further improving the models

In this section, we propose additional modifications of the models to improve the coreference resolution. First, in Section 4.4.1 we study the effect of previously mentioning the target item in the dialog and comparing it with the case the object is referred in the last turn.

Based on the manual error analysis (see Section 5.3), we observe that one of the common errors is that the models are unable to identify that the user is referring to a set of objects in the last turn, resulting in *under-prediction*, where the models predict fewer objects than they should. To address this issue, in 4.4.2 we propose to modify the models by adding an auxiliary task output head that predicts the number of references in the last user utterance.

To end this chapter, in Section 4.4.3 we experiment with combining different versions of the UNITER and BART-based models accordingly to their strengths in order to further enhance the overall performance.

4.4.1 Evaluating the 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 4.6 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, where the rest (43%) were new objects recently referred in the last user turn. Overall, UNITER and BART 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	Attributes	F1 score on mentioned	F1 score on 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 4.6 Performance of UNITER and BART-based models depending if the target object was previously mentioned in the dialog or not (new object). Both UNITER† and BART† are the best DSTC10 submissions adding multimodal context (prev. objs.) as part of the input. The systems may incorporate in the input object non-visual attributes (NV) or both non-visual and visual attributes (NV+V).

To sum up, we show that UNITER-based models are better on mentioned items, and the BART-based models are superior identifying new objects. This is corroborated in Section 4.4.2, and in Section 4.4.3 we propose a smart way of combining these models to increase the overall performance.

4.4.2 Addressing under-prediction

The described systems so far might 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 leads to systems that are *over-predicting*, i.e., the number of predicted referred objects is higher than the true number of referred objects or, analogously, *under-predicting*.

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 utterance, and this prediction will be used at inference time to modify the set of hypothesized referred objects. We implement this modification 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 (4.1).

$$\mathscr{L}_{\text{total}} = \lambda_{\text{mm-coref}}\mathscr{L}_{\text{mm-coref}} + \lambda_{\text{n-targets}}\mathscr{L}_{\text{n-targets}}. \tag{4.1}$$

$\mathscr{L}_{\text{n-targets}}$ 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. In the first approach, the N objects with highest coreference probability are selected if $N \leq 3$. If $N > 3$, only the 3 most probable objects are considered since we believe it is unlikely that 4 or more items are referred in the last turn, and then we can formulate the multi-class classification problem with 4 classes. The object probabilities of being referred are estimated using the output of the coreference head and the *softmax function* described in Equation (4.2), where o_i is the output of the coreference head of the i-th object out of K objects in the scene.

$$\text{softmax}(\mathbf{O})_i = \frac{e^{o_i}}{\sum_{j=1}^{K} e^{o_j}}. \tag{4.2}$$

This approach addresses both over and under-prediction, but it was not sufficiently effective to improve the performance of the models although the auxiliary head had 98% accuracy. The fact that the models were relying more on the prediction of the auxiliary head than on the actual predictions of the coreference head can be the cause why the model is not improving.

Taking that into account, an alternative strategy is tested which only addresses 'under-prediction'. During the error analysis phase (Section 5.3) it is noted that the models are struggling to identify all the referred objects even though it is clear that the user is referring to more items in the last utterance. Therefore, the auxiliary head prediction can be used when the model is under-predicting: whenever the auxiliary head reports a number of target objects higher than the actual number of predicted referred objects, the model is forced to return more items based on their probabilities up to the number of predicted targets by the auxiliary head. This second approach showed better results with both models. The details how this additional auxiliary head is implemented are described now.

Auxiliary head on the UNITER-based model

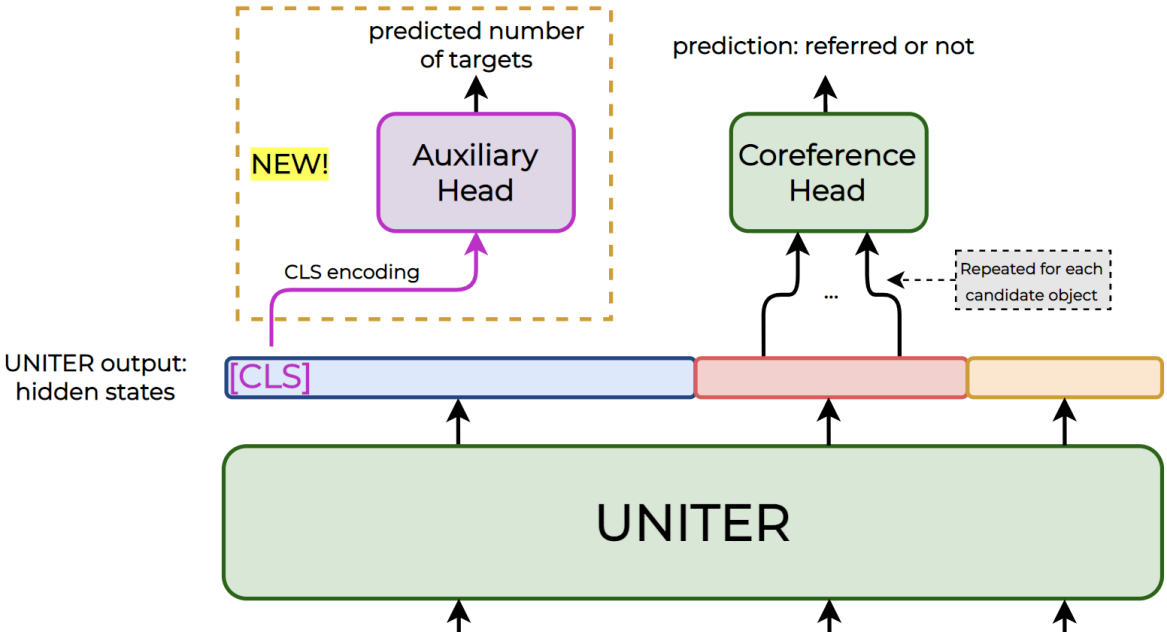


Fig. 4.6 Detailed diagram of the UNITER-based model proposed in Huang et al. (2021) including an additional auxiliary head that predicts the number of referred objects in the last user utterance. For the description of the rest of the model remember Fig. 3.3.

The special token [CLS] represents the whole input textual sequence of a transformer model (Devlin et al., 2019), and it is typically used for classification tasks. In this case, we use it for multi-class classification to determine the number of referred objects in the last user utterance. The output hidden state of UNITER that encodes the [CLS] token is fed into a single-layer neural network, called *Auxiliary Head*, that is trained to predict the number of target objects, i.e., the number of referred items in the last user utterance. Figure 4.6 represents the UNITER-based model after appending the new auxiliary head.

Auxiliary head on the BART-based model

The original BART-based system (Lee et al., 2021) does not use the typical special token [CLS] to summarize the natural language input. However, a special token <NOCOREF> was originally incorporated to identify the utterances without object references by using an additional auxiliary output head, called 'nocoref head'. Whenever this auxiliary head predicted no references, the predicted referred objects by the coreference head were overwritten and set to false. Since we showed this auxiliary head was not beneficial for the overall performance, we can re-use the special token <NOCOREF> to assess the auxiliary task of determining the number of referred objects by the user.

Figure 4.7 illustrates the detailed diagram of the BART-based system after incorporating the new Auxiliary Head that, as done with the UNITER model, aims to predict the number of target items modeling this task as a multi-class classification problem.

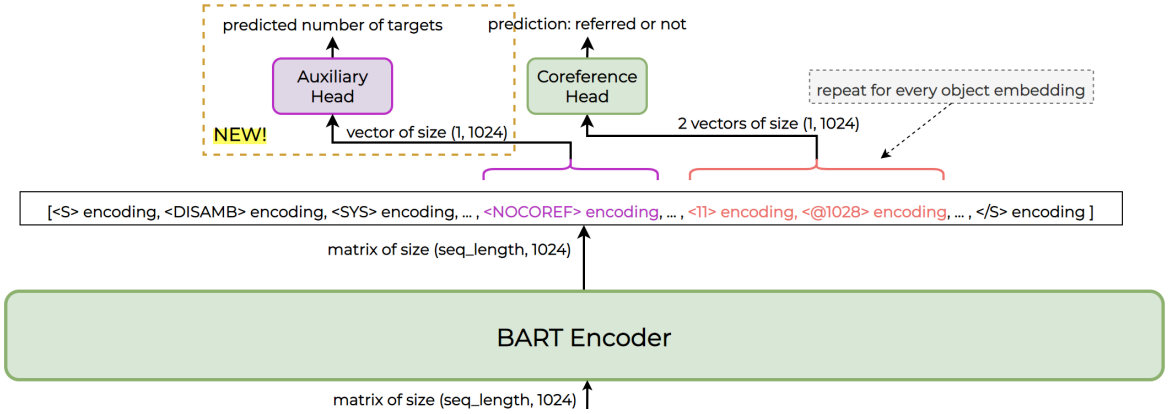


Fig. 4.7 Detailed diagram of the BART-based model proposed in Lee et al. (2021) including an additional auxiliary head that predicts the number of referred objects in the last user utterance. The rest of the model is detailed in Fig. 4.3.

Final results after implementing the auxiliary head

We compare the effect of adding the new auxiliary task head to the UNITER and BART based models and the results are shown in Table 4.7. After tuning the hyperparameters $\lambda_{\text{n-targets}}$ and $\lambda_{\text{mm-coref}}$ of Equation (4.1) all the models improved with respect its equivalent version without the additional task head. Appendix A contains more details about all the settings of the experiments and the hyperparameter tuning.

Base model	Attributes	Auxiliary head	F1 score on mentioned	F1 score on 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 4.7 Performance comparison of UNITER and BART-based models after incorporating the new auxiliary task head that predicts the number of referred objects by the user in his/her last dialog turn. The results are also reported splitting the set of objects depending if they were previously mentioned in the dialog (multimodal context) or not (new items). 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 $\sim$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.

4.4.3 Model combination

We observe in Tables 4.7 and 4.6 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.

Both models can be combined so 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 4.8 shows the results after combining several 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 since 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.**			**Legal on**	**F1 Score**
IDs	**Attrs.**	**Aux. head**	**IDs**	**Attrs.**	**Aux. head**	**DSTC10**	**overall**
	NV	✓	✓	NV	✓	✓	0.789
	NV	✓	✓	NV		✓	**0.800**
	NV	✓	✓	NV+V	✓		0.797
	NV	✓	✓	NV+V			**0.806**

Table 4.8 Results after combining UNITER and BART based models. The UNITER-based model operates on previously mentioned objects in the dialog, whether BART-based model just focuses on non-mentioned (new) items in previous utterances. It is compared the performance of different combined models depending if they are valid on DSTC10 competition. Additionally, there are BART-based systems that perform better on the whole set of objects, and others that are finer on the objects that were not previously mentioned.

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.

4.5 Discussion

Summarizing the results presented in this chapter, we have seen that the referring expressions on the furniture domain dialogs are easier to resolve than the fashion domain because there are fewer candidate objects. Moreover, it was shown in the cross-domain experiment that the UNITER-based model has more adaptability than the BART-based models due to the fact that it is not relying on the object IDs learned at training time. In contrast, the BART-based model is more robust if it is trained on the set of objects it is going to be used on, even though the object distributions, the physical stores or the object locations in the store are altered at inference time.

We have also shown that including object descriptions in the form of a list of attributes as part of the input of the models can boost the overall performance. In Huang et al. (2021) it is already shown that adding non-visual attributes as part of the object embeddings increases the performance for the UNITER-based model. We demonstrate the same for the BART-based model. Additionally, we also evaluate the effect of incorporating the visual attributes on the BART-based model, showing that it would result in additional boost in performance if we had available a perfect visual descriptor system at inference time.

After analyzing some common errors (in Section 5.3) we propose the addition of a new auxiliary task head at the output of the models to predict the number of referred items by the user in the last dialog turn. Using the output of this auxiliary head to force the models to identify more referred items turns out to be beneficial for the overall performance of both models: UNITER's performance increased by 0.3%, and BART's by 0.4% F1 score.

Finally, we observe that the UNITER-based model out-performs BART-based model on objects that were previously mentioned in the conversation, whether the BART-based model is superior than UNITER on identifying new objects just referred in the last turn. We combine the best models taking into account their strengths and the conversation context to achieve and beat the 0.80 F1 score mark.

Appendix A.1 contains detailed instructions for the experimental settings used in this chapter and on how models' loss weights are tuned.

Chapter 5

Human-level performance and Error Analysis

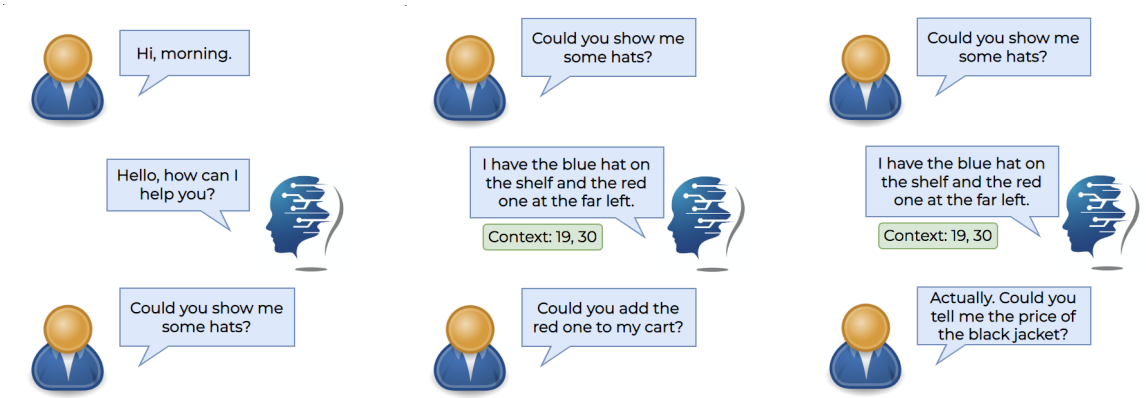
In order to calculate a potential upper bound for the performance of the models, in this chapter we estimate the human-level performance in Section 5.2, using a randomly sampled subset of the `devtest` set (Sec. 5.1). Furthermore, in Section 5.3 we show the most common errors of the models as well as the problems the annotators have found, what can provide insights into how further boost the systems.

5.1 Description of the devtest random subset

We have shown that the models can achieve a 0.80 F1 score on the coreference resolution task. In Section 5.2, we evaluate human performance on this task to estimate a hypothetical upper bound for the models' performance.

A set of 100 randomly selected examples from the `devtest` set is built. 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.

There are some examples that have no targets because the last user utterance is not referring to any particular object. In contrast, there are some examples with references. These referred items might have been previously mentioned in the dialog, so they are within the conversation context; or they may have just been referred in the last user turn. Examples of these three kinds of examples are showcased in Figure 5.1.



(a) No reference (b) In the conversation context (c) New referred object

Fig. 5.1 Comparing different types of referring expressions. The dialogs might have no references (left), they can be previously mentioned in the dialog (mid), or the object have just been referred in the last utterance (right). First, the user referred to a set of general objects; in the second example, it would be easy to identify that the referred item is the 'red one' with ID 30; and in the last example, the user talks about a new item not previously mentioned.

The proportions of this type of examples are calculated and illustrated in Table 5.1. The number of examples with no referred objects is similar between both sets. Furthermore, the proportion of targets within the conversation context is greater in the sampled random subset than the original SIMMC2 `devtest` set. We expect that both SOTA models and human annotators perform better on the previously mentioned items than in the recently referred objects, so the results are comparable. Note that the first statistic measures the proportion of examples, whether the second one counts the ratio of targets.

	SIMMC2 `devtest`	**Random subset**
Proportion of samples without targets	50%	53%
Proportion of targets in the conversation context	57%	75%

Table 5.1 Statistics of the 100-samples `devtest` random subset.

5.2 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.

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 5.2.

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 5.2 Estimating human performance and comparing it with different models. Evaluating human annotators on a random subset of 100 examples of the `devtest` set.

We observe that 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 5.3. We compute the Inter-Annotator Agreement by computing the Cohen's Kappa estimator (Cohen, 1960) using Equation (5.1).

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \tag{5.1}$$

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

As shown in Table 5.3, the estimated Cohen's κ mean is 0.856, which indicates high agreement between the annotators (Cohen, 1960).

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 5.3 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 (see Table 5.4). 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 Chapter 4 since the proportion of targets that are previously mentioned in the dialog happened to be higher in this test set (see Table 5.1). Although human evaluation was performed on a small dataset, the results suggests that the improved models may already achieve human-level performance.

Base model	IDs	Attributes	Auxiliary head	F1 score on mentioned	F1 score on new objs.	F1 score overall
UNITER†	✓	NV		0.877	0.649	0.821
UNITER†	✓	NV	✓	0.911	0.595	0.832
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 5.4 Estimating human performance and comparing it with different models. Evaluating models on random subset of 100 examples.

5.3 Error analysis

We manually examine errors made by the models and by the human annotators. In general, the models tend to struggle resolving coreferences in these scenarios:

- *Under-prediction*: the models are predicting fewer referred items than expected although it is obvious the user is referring to a larger set of objects in the last utterance.
- *Over-prediction*: the models are predicting more referred objects than they should.
- *Unrecognized location*: the models struggle to identify the object/s referred using a location expression, such as "on the right of something", "the second on the top", etc.
- *Insufficient focus on the conversation context*: the objects are recognizable by focusing on the conversation context, but the models tend to rely on other features.
- *Annotation errors*: some examples show that the SIMMC2 dataset is not completely clean and has some annotation errors.

- *Improvable dataset and task design*: we illustrate some examples where the dataset can improve in future versions.

All the scenarios are illustrated below with error examples taken from the models studied in Chapter 4. In the images, the objects are marked in red when they are ground-truth targets, in green when they are model's predictions, or in white when they are previously mentioned in the conversation context.

Models are under-predicting

Figure 5.2 shows two examples where the models are not able to recognize that there are some referred items although clear referring expressions are used, like "that grey option", "the black option you showed me", "the green one in the middle". 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. The threshold cannot be diminished in all cases since it would produce more false positives, affecting negatively to the overall F1 score. In Chapter 4 (Sec. 4.4) we describe how we addressed this error by predicting the number of objects and applying heuristics that forced the model to predict more items.

Models are over-predicting

Analogously, the models can predict as referred more items than they are expected to do so. In Figure 5.3 we can observe that this is a common mistake if there are several objects with the same properties. In the first example, several jeans are shown in the store's cubby, but just two of them are the true targets. In the same way, various military-like trousers are predicted as referred but the user just focused on a subset of them. 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.

Moreover, Figure 5.3 exemplifies the difficulty of handling referring expressions based on locations. 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 cubby", "two cubbies over" or "on either side of the grey trousers". 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.

DIALOGUE HISTORY of example No. 1731

User : Could you recommend a coat for me? System : There is a black coat in the front on the right, a grey option in the section just behind it, and a light grey option on the left in the second to last section. <SOM> <6>, <12>, <7> <EOM> User : I'll take that grey option, as well as the black option you showed me.

(a) There are two referred coats that are not identified by the model, marked in red squares.

DIALOGUE HISTORY of example No. 4921

System : What do you think about the black hat on the right with the red logo, or the green camo hat, or the green hat next to it? <SOM> <70>, <73>, <72> <EOM> User : What's the price on the black one and the green one? System : Which hat are you referring to? <SOM> <EOM> User : That black one you suggested and the green one in the middle.

(b) There are two hats not predicted by the system, marked in red.

Fig. 5.2 Error examples of cases where the models are predicting fewer referred items than they should. They models should be able to infer that there are some referred items just by focusing on the last user turn.

DIALOGUE HISTORY of example No. 5508

System : Ok. I will add that hoodie now. <SOM> <6> <EOM> User : Can you tell me the ratings of the two pairs of jeans? System : Which ones? <SOM> <EOM> User : The pair of jeans in the rightmost cubby and the pair two cubbies over.

(a) All jeans are predicted as referred because they have common descriptions and the models fail to identify that just a subset is the correct target.

DIALOGUE HISTORY of example No. 4173

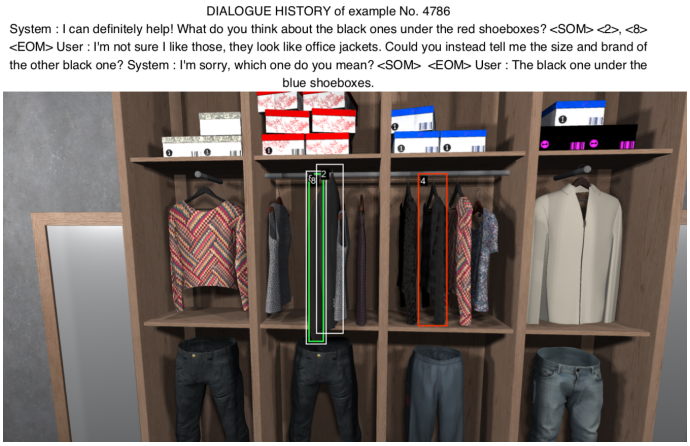
System : It is in size XL. <SOM> <7> <EOM> User : What other items do you have that you recommend? System : There is a pair of grey trousers amongst the green trousers . <SOM> <28> <EOM> User : Tell me how the green trousers compare. The ones on either side of the grey trousers.

(b) All military-like trousers are predicted as referred because the models fail to identify that just a two of them are the ground-truth targets.

Fig. 5.3 Error examples of cases where the models are predicting more referred items than they are expected.

Models are struggling with locations

Some referring expressions may be using relative locations to ground the mentioned objects, such as the first example in Figure 5.4. Here, the target object is referred using the expression "under the blue shoeboxes", so the object of interest has to be identified by first knowing what is a shoebox, and then recognize that the referred object is located under it.



(a) The models are incapable of identifying the "blue shoeboxes" since they are not described in SIMMC2 meta-files nor pre-trained on them.

(b) Several objects are mentioned in the conversation context and the models struggle to identify

Fig. 5.4 Error examples where models are struggling with relative locations to ground the mentioned items.

In the second example of Fig. 5.4, several objects are mentioned in the conversation. Just two of them are the ones referred in the last turn, but the model has to identify which are them by also resolving their locations in the previous utterance.

Therefore, the coreference resolution systems has to be able to recognize not only the set of objects of interest (fashion and furniture items on SIMMC2) but also all the objects in the scene since the referred objects can be ground using expressions based on other objects. Additionally, the models should be pre-trained or fine-tuned on identifying referring expressions based on locations, such as "on top of something", "on the left...", etc. but this is not a simple task since these expressions can be relative.

Models do not focus enough on the conversation context

Some examples indicate that the models are not weighting enough the conversation context, i.e., the systems are not relying on the set of objects previously mentioned. Figure 5.5 shows an example where the user clearly refers to the last mentioned items but the model still fails to predict them. We believe that the overall performance of the models can be increased if these type of examples are fixed just by focusing on the conversation context. This is not an easy task since excessively relying on the conversation context can lead to degrading the performance on objects not mentioned before in the dialog.

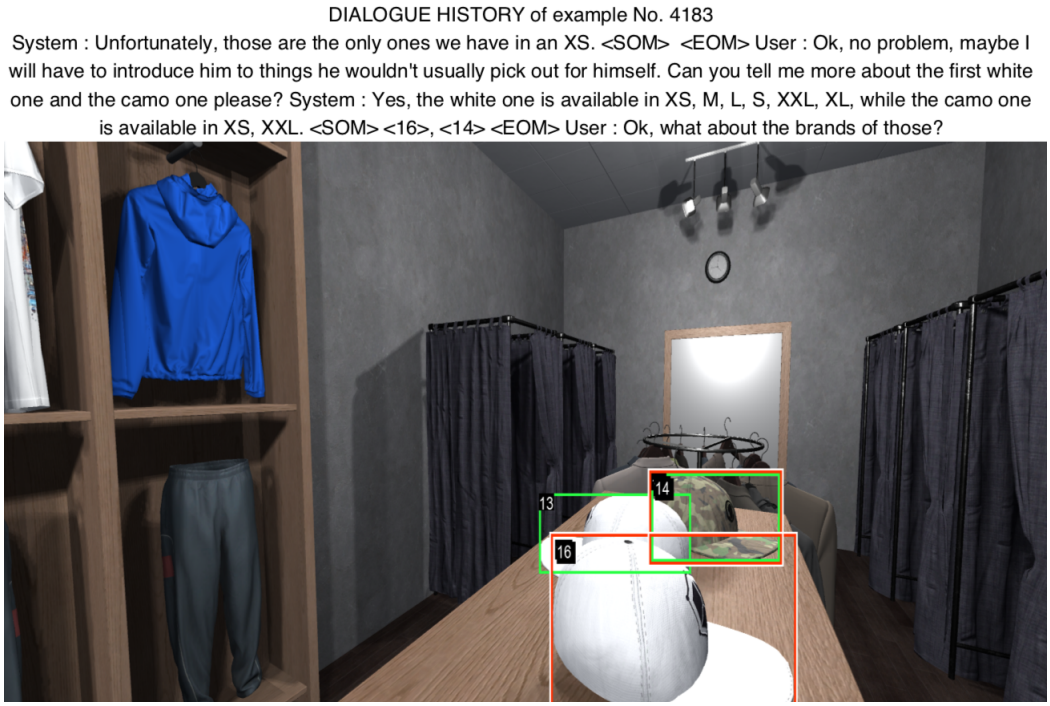


Fig. 5.5 Error examples where the systems do not use the conversation context enough.

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 5.6 showcases two examples where

the targets are missing when they clearly should be present, or even the dialog does not correspond with the annotated targets. For example, in the first image we can check that the user is referring the the previously mentioned items, but the annotations just contain one single target. Moreover, the second image illustrates an example where the user is clearly referring to the area rug, but the annotations do not consider any target.

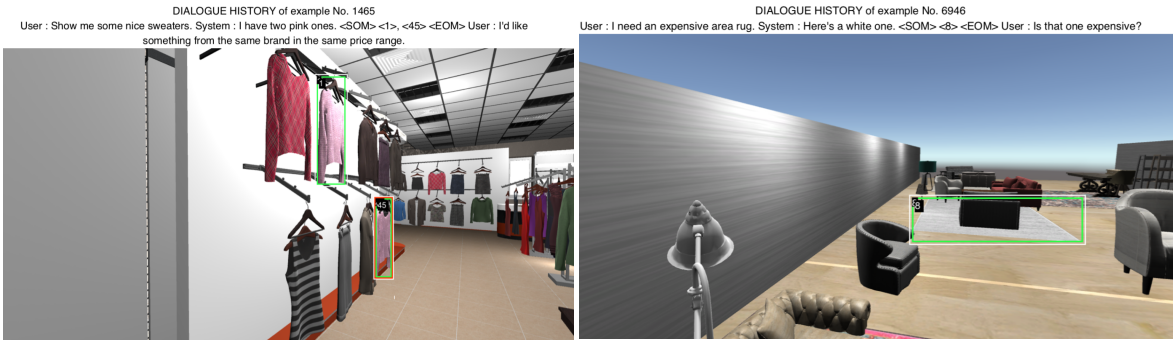


Fig. 5.6 Examples where the dialogs are wrongly annotated.

Improvable dataset and task design

We also perceive additional patterns in the error examples. First, Figure 5.7 shows one pitfall of the SIMMC2 dataset: it has fixed scenes, so the camera does not move. In a real scenario, if the referred object is not close or visible, the assistant and the user would move next to it. In this dataset, the scenes are still, so system might have problems to visually identify some items.

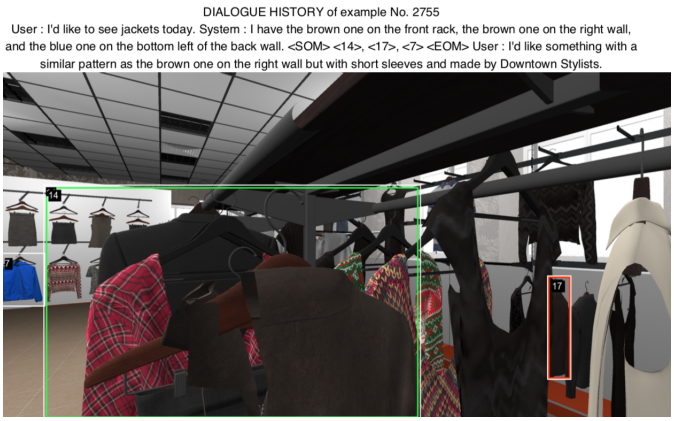


Fig. 5.7 An error example where the referred object is not clearly visible from the current camera angle.

Another drawback of the approaches of the models is that they require the use of a fixed dialog window, since they cannot process enormous amount of text at once. In Figure 5.8

we can see an example where there is no information about the referred item since it was mentioned several turns ago and the dialog window does not capture the crucial utterance.

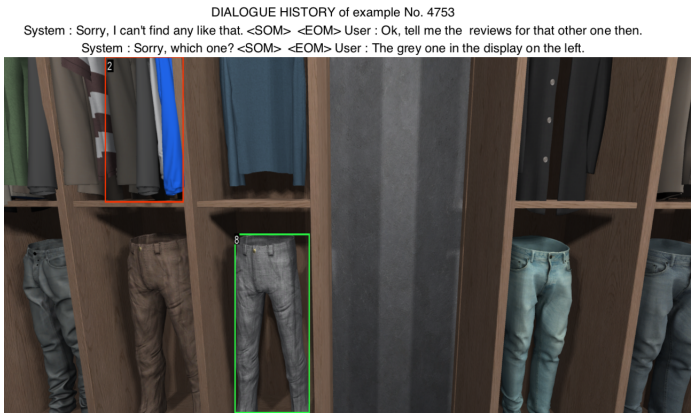


Fig. 5.8 An error example where dialog context is so short that is impossible to know the referred object, since it was mentioned several turns ago.

5.3.1 Human annotators errors

In Section 5.2 we describe a human annotation task where the annotators are asked to analyze some random examples from the `devtest` set and identify the referred objects. Then, the annotations are evaluated and the human-level performance is estimated. In this section we study the common errors of the human annotators.

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, also ratifying the Inter-Annotator Agreement estimation computed in Section 5.2. Here, we analyze some examples wrongly annotated by all the humans. In the following images, target objects are marked in red and predictions made by humans in green.

The first source of errors comes from the fact that SIMMC2 is not completely clean. All human annotators failed to correctly classify the examples illustrated in Fig. 5.9 because we believe this example is annotated incorrectly. As we can see, the last user utterance asks for two white rags. Human annotators mark the two whitish rags in the scene, but the SIMMC2 ground-truth target is just the big rag of the middle.

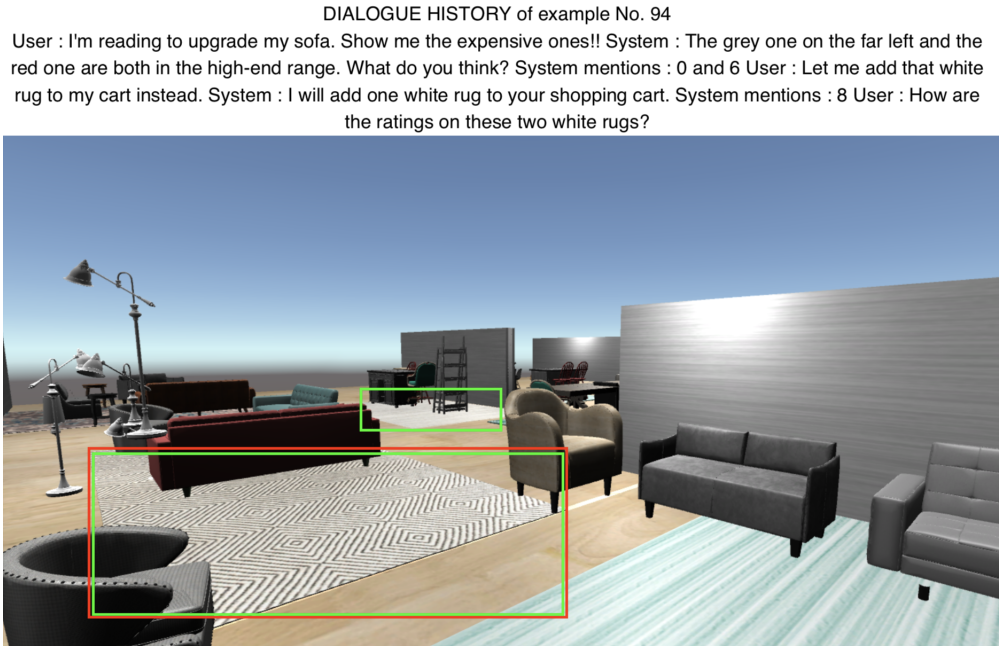


Fig. 5.9 An error example where SIMMC2 is wrongly annotated and humans predicted the two most probable rags accordingly the dialog and the scene.

On the other hand, there are examples that are correctly classified, but they are just hard to resolve. For example, Figure 5.10 shows a scene with multiple objects matching the description provided in the dialog. The conversation refers to a couple of black coats hanging on two racks on the left. However, there are multiple racks and several black clothing items, making the task of resolving the referred objects really tough.

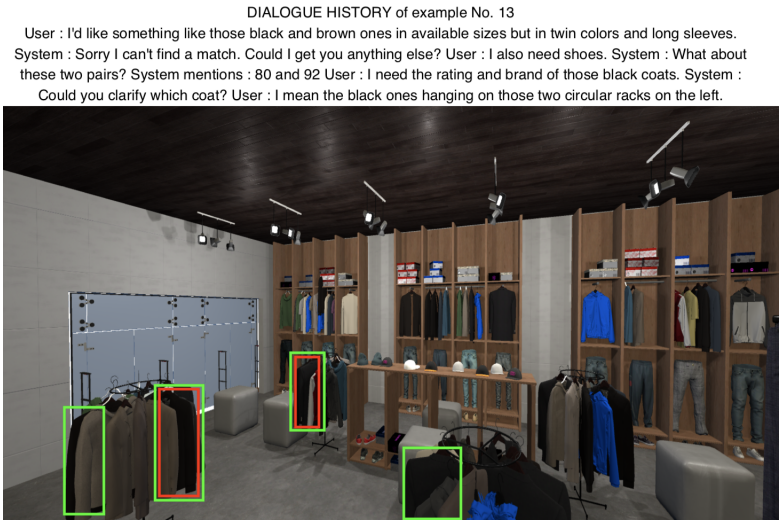


Fig. 5.10 An error example where the annotators fails to identify correctly all referred objects since there are multiple options that match the description provided in the dialog.

Similarly, humans tend to struggle differentiating colors. This is exemplified in Figure 5.11. It is clear that 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". This would give the BART-based model an advantage compared to humans since it is training the embedder using the ground-truth provided in the metadata files. Moreover, the other referred object is a grey coat, but there are plenty of items matching that description in the upper shelf. Like in the previous Figure 5.10, annotators failed to recognize the target item because of the range of feasible possibilities in the scene.

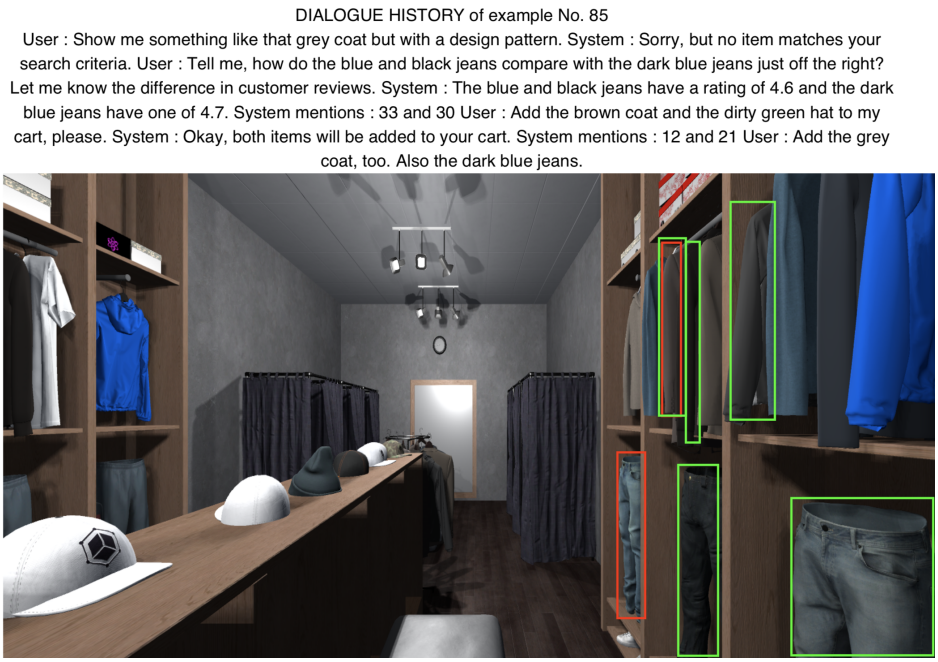


Fig. 5.11 An error example where the annotators fails to identify correctly all referred objects since there are multiple options that match the description provided in the dialog and humans struggle to distinguish similar colors.

5.3.2 Analysis summary

We have seen that several examples are hard to tackle since there are many object 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. This error report is used to refine the models throughout the project improving the overall performance (Chapter 4). The current chapter can also be employed as a source of inspiration for future work.

Chapter 6

Conclusion and Future Work

The main goal of this thesis is to study, analyze and improve state-of-the-art multimodal coreference resolution systems basing our research on the proposed systems for the DSTC10 challenge and using the SIMMC2 dataset (Kottur et al., 2021).

We first replicate the two best performing models of the DSTC10 competition: the UNITER-based model introduced in Huang et al. (2021) and the BART-based model detailed in Lee et al. (2021). We also propose various improvements to the replicated systems, 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 the other specific tasks.

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

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, achieving the best performing results of the project, beating the DSTC10 top score of 0.743, reaching 0.800 F1 score.

We are interested to compare models performance with the human-level performance on the coreference resolution task on the same dataset. To do so, three human annotators are asked to classify 100 random examples from the `devtest` set. The results exhibit that the models are already performing similar to the human-level, making it difficult to intuitively enhance them.

Finally, an extensive error analysis is carried out to understand the behavior of the models and gain insights to further enhance them. This is the case of some error cases, that turned out to be crucial to propose the new auxiliary task head that brought some rewarding modifications. The error analysis can be the starting point of future directions.

The project is open sourced and can be found at the following GitHub repository: https://github.com/AlejandroSantorum/simmc2-Multimodal_Coreference_Resolution.

6.1 Future work

There are several experimental directions that can be followed in order to keep improving on the multimodal coreference resolution task.

First, there is always the necessity of adaptable models that generalize several domains even though the models were not trained on them. Zero-shot and few-shot learning techniques could be explored to boost the models.

On the other hand, we note that the visual features are not sufficiently used. Scene images do not seem to provide useful information to the systems. The poor quality of some SIMMC2 scenes can be the core of this issue, and it could be fixed by a version of this dataset with multiple images per scene from different view points.

Furthermore, the BART-based model benefits from considering attributes as part of the input. However, we believe the mechanism detailed in Figure 4.3, where the attributes embeddings are combined with the object embeddings by summing both of them, can be optimized.

Similarly, we could further investigate and improve the heuristic employed at inference time when including the additional task head. Right now the head is just focusing the models' under-predicting trend, but we suggest it could be refined.

Finally, other common errors can be addressed after considering the error analysis of Section 5.3. Error analysis has already been the source of inspiration for some successful improvements, so fixing other usual errors might bring advanced modifications. In particular, references about background objects (racks, cubbies, shelves, ...) can be addressed in the future.

References

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8578734.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.* https://dl.acm.org/doi/pdf/10.5555/944919.944966.

Bergsma, S. and Lin, D. (2006). Bootstrapping path-based pronoun resolution. *Association for Computational Linguistics (ACL)*. https://aclanthology.org/P06-1005.pdf.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *Computing Research Repository (CoRR)*. https://arxiv.org/pdf/2005.14165.pdf.

Chen, Y., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2020). UNITER: Learning universal image-text representations. *European Conference on Computer Vision (ECCV)*. https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123750103.pdf.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*. https://journals.sagepub.com/doi/10.1177/001316446002000104.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Association for Computational Linguistics*. https://aclanthology.org/N19-1423.pdf.

Huang, Y., Wang, Y., and Tam, Y. (2021). UNITER-based Situated Coreference Resolution with rich multimodal input. *Computing Research Repository (CoRR)*. https://arxiv.org/abs/2112.03521.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *Computing Research Repository (CoRR)*. https://arxiv.org/pdf/2001.08361.pdf.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR)*. https://arxiv.org/pdf/1412.6980.pdf.

Kottur, S., Moon, S., Geramifard, A., and Damavandi, B. (2021). SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. *Association for Computational Linguistics (ACL)*. https://aclanthology.org/2021.emnlp-main.401.pdf.

Kottur, S., Moura, J. M. F., Parikh, D., Batra, D., and Rohrbach, M. (2018). Visual coreference resolution in visual dialog using neural module networks. *European Conference on Computer Vision (ECCV)*. https://openaccess.thecvf.com/content_ECCV_2018/papers/Satwik_Kottur_Visual_Coreference_Resolution_ECCV_2018_paper.pdf.

Kumar, A., Aurisano, J., Di Eugenio, B., Johnson, A., Alsaiari, A., Flowers, N., Gonzalez Martinez, A., and Leigh, J. (2017). Multimodal coreference resolution for exploratory data visualization dialogue: Context-based annotation and gesture identification. *Workshop on the Semantics and Pragmatics of Dialogue*. http://semdial.org/anthology/Z17-Kumar_semdial_0008.pdf.

Lee, H., Kwon, O. J., Choi, Y., Kim, J., Lee, Y., Han, R., Kim, Y., Park, M., Lee, K., Shin, H., and Kim, K.-E. (2021). Tackling situated multi-modal task-oriented dialogs with a single transformer model. *Association for Computational Linguistics (ACL)*. https://openreview.net/forum?id=NajekV9uBas.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Association for Computational Linguistics (ACL)*. https://aclanthology.org/2020.acl-main.703.pdf.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. *European Conference on Computer Vision (ECCV)*. https://arxiv.org/pdf/2004.06165.pdf.

Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P. (2017). Focal loss for dense object detection. *IEEE International Conference on Computer Vision (ICCV)*. https://openaccess.thecvf.com/content_ICCV_2017/papers/Lin_Focal_Loss_for_ICCV_2017_paper.pdf.

Loshchilov, I. and Hutter, F. (2018). Fixing weight decay regularization in adam. *ICLR 2018 Conference Blind Submission*. https://openreview.net/forum?id=rk6qdGgCZ.

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. *Computing Research Repository (CoRR)*. https://arxiv.org/pdf/1806.08730.pdf.

Moon, S., Kottur, S., Crook, P. A., De, A., Poddar, S., Levin, T., Whitney, D., Difranco, D., Beirami, A., Cho, E., Subba, R., and Geramifard, A. (2020). Situated and interactive multimodal conversations. *Computing Research Repository (CoRR)*. https://arxiv.org/pdf/2006.01460.pdf.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *Proceedings of the 38 th International Conference on Machine Learning*. http://proceedings.mlr.press/v139/radford21a/radford21a.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *Computing Research Repository (CoRR)*. https://arxiv.org/pdf/1910.10683.pdf.

Ramanathan, V., Joulin, A., Liang, P., and Fei-Fei, L. (2014). Linking people in videos with "their" names using coreference resolution. *European Conference on Computer Vision (ECCV)*. http://vision.stanford.edu/pdf/vignesh14.pdf.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. *Association for Computational Linguistics*. https://aclanthology.org/D19-1410.pdf.

Seo, P. H., Lehrmann, A. M., Han, B., and Sigal, L. (2017). Visual reference resolution using attention memory for visual dialog. *Neural Information Processing Systems (NIPS)*. https://arxiv.org/pdf/1709.07992.pdf.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Association for Computational Linguistics (ACL)*. https://aclanthology.org/J01-4004.pdf.

Tan, H. and Bansal, M. (2019). LXMERT: Learning cross-modality encoder representations from transformers. *Association for Computational Linguistics (ACL)*. https://aclanthology.org/D19-1514.pdf.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is aal you need. *Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wen, T.-H. and Young, S. (2020). Recurrent neural network language generation for spoken dialogue systems. *Computer Speech & Language*. https://www.sciencedirect.com/science/article/pii/S0885230817300578.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Yejin, C., and Gao, J. (2021). VinVL: Revisiting visual representations in vision-language models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9577951.

Zheng, Q., Diao, X., Cao, J., Zhou, X., Liu, Y., and Li, H. (2018). Multi-modal coreference resolution with the correlation between space structures. *Computing Research Repository (CoRR)*. https://arxiv.org/pdf/1804.08010.pdf.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*. https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Zhu_Aligning_Books_and_ICCV_2015_paper.pdf.

Appendix A

Experimental settings

In this chapter we exhibit the settings to replicate all the described experiments in Chapter 4. In Section A.1 we elaborate on the details of the experiments, and in Section A.2 we specify how the hyperparameters were tuned.

The open source code of this project is publicly available in the GitHub repository: **https://github.com/AlejandroSantorum/simmc2-Multimodal_Coreference_Resolution**.

A.1 Experiments details

A.1.1 Models replication

The UNITER-based model (Huang et al., 2021) and the BART-based model (Lee et al., 2021) can be replicated as follows:

UNITER-based model:
The overall system is trained using focal loss with $\gamma = 2$ and $\alpha = 1$ for the negative class (when an object is not referred), and $\alpha = 5$ for the positive class (the object is referred). It is used the Adam optimizer with a learning rate of $5 \cdot 10^{-6}$ and $\varepsilon = 10^{-8}$, and a batch size of 16. Moreover, the model is trained for a maximum of 30 epochs and performed early-stopping as a function of the F1 score on the SIMMC2.0 `dev` set.

BART-based model:
The 24-layer BART Transformer model is fine-tuned for 10 epochs with an initial learning rate of $5 \cdot 10^{-5}$ and a batch size of 16 with AdamW optimizer. The best checkpoint evaluated at every 1000 steps is chosen on the `devtest` set. The joint learning coefficients are:
$\lambda_{\text{LM}} = 1.0$; $\lambda_{\text{mm-coref}} = 0.8$; $\lambda_{\text{retrieval}} = 0.4$; $\lambda_{\text{mm-disamb}} = \lambda_{\text{att}} = \lambda_{\text{empty-coref}} = 0.1$

Once the BART-based model is replicated, the task heads that are not tackling coreference resolution are suppressed, and the considered $\lambda_{\text{mm-coref}}$ is $\lambda_{\text{mm-coref}} = 3$. This is not important since only $\mathscr{L}_{\text{mm-coref}}$ loss remains.

A.1.2 Individual evaluation on each SIMMC2 domain

The models are trained as in the replication experiments. We split the `devtest` into two subsets depending on the dialog domain. Finally, the models are evaluated on each split subset to get the performance on each individual domain.

A.1.3 Adding non-visual and visual attributes to the input of BART

The BART-based model is trained and evaluated using the default parameters (replication instructions) but the *training batch size* is reduced from 16 to **8** to account for the extra features at the input. Additionally, the model is fine-tuned for **12 epochs** instead of 10.

A.1.4 Cross-domain and cross-scene evaluation

The settings of the UNITER-based model are unaltered from the replication experiment. Only take into account the considered datasets (for both training and testing) are changed depending on the experiment (see Table 4.3).

For the BART-based model, the default settings are unchanged but the training batch size, that is reduced from 16 to **8**, and the number of epochs is now reduced from 10 to **8** since the datasets are smaller than the standard SIMMC2 training set.

A.1.5 Evaluating the effect of the reference type

The models are trained using the default parameters, or using the instructions provided in previous sections if we are considering an modified model (e.g., adding non-visual and visual attributes to the input of the BART-based model).
The predictions of each model are compared to the multimodal context of each test example to split both the targets and the predictions depending if they were previously mentioned in the conversation. Lastly, the F1 score is calculated on each disjoint set (mentioned vs new target objects).

A.1.6 Adding the auxiliary task output head

The settings of the UNITER-based model are unaltered from the replication experiment but the weights of the losses. When adding a new auxiliary task output head, we are also incorporating an additional loss. After tuning the weights of the losses (see Sec. A.2 for more details), we best configuration is $\lambda_{\text{mm-coref}} = 0.75$ and $\lambda_{\text{n-targets}} = 0.25$, where $\lambda_{\text{n-targets}}$ is the weight of the auxiliary head.
With the BART-based model we keep using a training batch size of 8 and training for 12 epochs. Also, after tuning the weights of the losses (see Sec. A.2 for more details), the best configuration is $\lambda_{\text{mm-coref}} = 6$ and $\lambda_{\text{n-targets}} = 1$.

A.1.7 Experiments of model combination

The considered models are trained accordingly the previous sections. To combine the models, the set of targets and the set of predictions are split depending if the object was previously mentioned in the conversation or not. Then, we models are evaluated individually on each corresponding subset.

A.2 Hyperparameter tuning

Machine learning models usually need to tune a set of special parameters depending on the architecture of the model or on the learning algorithm, such as number on neurons, number of hidden layers, learning rates, loss weights, etc. Different configurations are checked and the performance is evaluated on a validation set or, in this project, on the `dev` set.

In this project, we considered the default hyperparameters of the models proposed by the original authors (Huang et al., 2021; Kottur et al., 2021; Lee et al., 2021), so the results could be comparable. However, when we modified the models to attach a new *auxiliary task output head*, we needed to optimize the trade-off between the weights of the coreference head and the new auxiliary head.

A.2.1 Considered weights for the UNITER-based model

Several pairs of $(\lambda_{\text{mm-coref}}, \lambda_{\text{n-targets}})$ we investigated as they are shown in Table A.1.

$\lambda_{\text{mm-coref}}$	0.97	0.95	0.9	0.85	0.83	0.8	0.75	0.7
$\lambda_{\text{n-targets}}$	0.03	0.05	0.1	0.15	0.17	0.2	0.25	0.3

Table A.1 Loss weight tuning for the UNITER-based model

The best performing configuration resulted to be $(\lambda_{\text{mm-coref}}, \lambda_{\text{n-targets}}) = (0.75, 0.25)$.

A.2.2 Considered weights for the BART-based model

Table A.2 contains the pairs of $(\lambda_{\text{mm-coref}}, \lambda_{\text{n-targets}})$ that were investigated.

$\lambda_{\text{mm-coref}}$	3	5	6	8	10	12
$\lambda_{\text{n-targets}}$	1	1	1	1	1	1

Table A.2 Loss weight tuning for the BART-based model

Three different BART-based models were tuned:

- The considered configuration for the BART-based model using coreference head and auxiliary task head was $(\lambda_{\text{mm-coref}}, \lambda_{\text{n-targets}}) = (3, 1)$.
- The configuration for the BART-based model using coreference and auxiliary task heads including textual non-visual attributes in the input was $(\lambda_{\text{mm-coref}}, \lambda_{\text{n-targets}}) = (12, 1)$.
- The best performing configuration for the BART-based model using both coreference and auxiliary task heads including textual visual and non-visual attributes in the input was $(\lambda_{\text{mm-coref}}, \lambda_{\text{n-targets}}) = (6, 1)$.