Deliverable D2.6: Semantics-based localisation in relevant environments

Due Date: 30/11/2022

Main Author: HWU

Contributors: CVUT, HWU

Dissemination: Public Deliverable

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

DOCUMENT FACTSHEET

Deliverable	D2.6: Semantics-based localisation in relevant environments
Responsible Partner	HWU
Work Package	WP2: Environment Mapping, Self-localisation and Simulation
Task	T2.3: Language-Driven Semantic Localisation
Version & Date	Final 30/11/2022
Dissemination	Public Deliverable

CONTRIBUTORS AND HISTORY

Version	Editor	Date	Change Log
1	HWU	03/11/2022	First Draft
2	CVUT	30/11/2022	Reviewed by CVUT
3	HWU	30/11/2022	Final version

APPROVALS

Authors/editors	HWU: Jose L. Part, Chirstian Dondrup, Daniel Hernandez Garcia, Oliver Lemon, CVUT: Rakshith Madhavan, Martin Zderadicka, Tomas Pajdla
Task Leader	CVUT
WP Leader	CVUT

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Contents

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
ARI	Social assistive robot used by the SPRING project
BIU	Bar-Ilan University (SPRING Partner)
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CVUT	Czech technical university in Prague (SPRING Partner)
HRI	Human Robot Interaction
HWU	Heriot-Watt University (SPRING Partner)
IGG	Image-Grounded Conversations
INRIA	Institut National de Recherche en sciences et technologies du numérique (SPRING Partner)
LSTM	Long short-term memory
MPC	Multi Party Conversation
PAL	PAL Robotics (SPRING Partner)
PNP	Petri-Net Planner
RelTR	Relation Transformer
ROS	Robot Operating System
SLAM	Simultaneous Localization And Mapping
SPRING	Socially Pertinent Robots in Gerontological Healthcare
UNITN	University of Trento (SPRING Partner)
VQA	Visual Question Answering
WP	Work Package (of the SPRING project)
YOLACT	You Only Look At CoefficientTs

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Executive Summary

Deliverable 2.6 reports on the visual semantics modules. This report delivers (i) software with a Natural Language vocabulary (categories) for object recognition that interfaces with the high-level task scheduler for semantic scene understanding (T2.3) and builds towards interfacing the Interaction Manager to incorporate object affordances (T2.4), and (ii) updated software for object detection and localization in 3D map based on semantic segmentation and unknown object detection (T2.3) integrated in ARI framework (T7.4) Currently, data used to test the methods and software were obtained in relevant realistic laboratory environments. Further testing in hospital environments integrated with user tasks is foreseen for the future as it depends on the advancement in the development of user cases.

This project has received funding from the European Union's Horizon 2020
Research and Innovation Programme under Grant Agreement No. 871245.

1 Introduction

The ability to understand and talk about their surroundings is of utmost importance for assistive robots. This is not only relevant for successfully performing tasks but also to answer questions and have situated dialogues that are coherent with respect to both the location where the dialogue is taking place as well as previous dialogue turns. 'Visual language grounding' refers to the connection between words (symbols) and their referents in the visual space, e.g., objects, their relationships, their attributes, etc.

Visually grounded dialogue for artificial systems is a challenging problem for many reasons. First of all, how do we ground language onto the visual world? Languages are very rich and the same words can mean different things in different contexts or to different people. Moreover, languages can be dynamic, i.e., the meanings of words evolve over time and new words are incorporated into the language. Meanings can also be created and negotiated dynamically by agents collaborating on a task. Therefore, it is unreasonable to encode all of this knowledge beforehand. Secondly, assuming the artificial agent, e.g., a robot, is able to understand language and ground it onto the visual world, how does it answer questions? In other words, how does it generate coherent and relevant responses given a scene, a dialogue history, and a question?

This document reports on the software modules for visual based semantic scene understanding. Chapter 2 introduces some of the most popular tasks related to visual dialogue, including language grounding, and scene graph generation. Chapters 3 and 4 introduce the developments for semantic segmentation and unknown object detection and localization in 3D maps. Chapter 5 presents the semantic scene graphs for visual grounded dialogue.

The software will be released in the [33, 34] code repositories. As per European Commission requirements, the repository will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software to the project coordinator at `spring-coord@inria.fr`.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

2 Related work: Vision and Language Research

Being able to have natural conversations about the space where they operate is very important for robots that need to interact with human users. This ability requires a deep understanding of the surrounding space and the dynamics of the environment. There are many tasks that have been proposed to attempt to fully or partially address this problem, e.g., image and video captioning, Visual Question Answering (VQA), Visual Dialogue, etc. For example, VQA [6] consists of producing an appropriate open-ended response given an image and a question. Solving this task requires not only an extensive understanding of the scene but also access to common-sense knowledge. Antol et al. [6] proposed two paradigms for evaluating such systems. In the first case, the system is tasked with producing free-form responses whereas in the second case, the task is posed as answering multiple choice questions. However, in neither case is the system required to generate the answers. Systems are usually given a set of human-generated answers to choose from. The difference is that in the multiple choice scenario, the choices belong to a small set of 18 answers, whereas for the open-ended scenario the answers are chosen from a larger pool of answers (generally 1000), which are selected according to various criteria. This has the aim to facilitate evaluation and simplify the task but it also limits the kind of responses the system is able to produce and thus, given that the output is a probability distribution over available answers, it is unclear what the use of such system could be for Human-Robot Interaction (HRI).

Visual Dialogue [11] is a task devised to evaluate a system's ability to hold conversations grounded on an image. In principle, given an image, a dialogue history and a question, the system is tasked with coming up with an accurate answer. As opposed to VQA, which focuses on one-shot interactions, visual dialogue requires the system to keep track of previous turns, conditioning future answers on the conversation history. The way performance is evaluated is based on Mean-Reciprocal-Rank (MRR), i.e., given a sorted list of candidate answers, the position of the ground-truth is retrieved and the mean of the inverse of this position is calculated for all question-answer pairs in the dataset:

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i} \tag{2.1}$$

where $rank_i$ is the position of the ground-truth response to question q_i and Q is the number of questions in the evaluation set.

At every turn in the dialogue during evaluation, the system is given an image, the *ground-truth* dialogue history up to the point that is being evaluated, a question, and a list of 100 answers that need to be ranked.

However, real-world dialogue is somewhat more complicated than this. First of all, we don't have a set of possible predefined answers to choose from. Second, we don't have access to the "ground-truth" of our conversation but rather, the history of the dialogue we actually had, with "mistakes" and all. And lastly, given that languages are very rich, there can be a large number of ways to correctly answer the same question.

The Audio-Visual Scene-Aware Dialog task [2] augments the Visual Dialogue task [11] with audio and video data. The goal in this case is to ground the dialogue on a sequence of frames, allowing to generalise the idea of visual dialogue to discuss events that happen over a period of time. As was the case for the previous task, the system is given a list of 100 answers to choose from at every dialogue turn.

The Visual Genome [19] is a large dataset of images annotated with objects, their attributes and relationships to other objects, natural language descriptions, and question/answer pairs. Moreover, every image is also represented by a scene graph that encodes some of this information in a structured representation format. However, the question/answer pairs are presented in isolation and as such, do not contain dialogue annotations.

The GuessWhat? [12] task is a cooperative game where a questioner is tasked with finding an object in an image by asking *yes-no* questions to an oracle. Besides identifying the correct object, the questioner is encouraged to ask informative questions by being penalised for every question asked. In other words, the goal is to identify the target object with the least number of questions. The main objective of this task is to endow machines with higher-level reasoning capabilities to both understand natural language descriptions and ground them on the visual world. However, since the oracle is only required to provide binary answers (yes/no), the resulting dialogues are quite limited and do not represent natural open-ended dialogue. Moreover, since the evaluation is only based on whether the correct object was identified or not, it is not clear what representations such models are learning in order to successfully complete

Figure 2.1: Example of conversations from IGC (left) and Visual Dialogue (right). Image reproduced from [26].

the task. To address the latter, Suglia et al. [36] proposed a multi-task evaluation framework which has the aim to encourage the learning of compositional representations. Concretely, apart from task success, they also assess the model's ability to predict attributes as well as its generalisation capabilities by mean of a zero-shot evaluation.

Mostafazadeh et al. [26] proposed the Image-Grounded Conversations (IGC) task, which resembles Visual Dialogue [11] but is motivated by a different goal. Whereas in Visual Dialogue, the goal is to have a conversation about an image, IGC aims to have open-ended conversations that, while grounded on an image, can go beyond what's depicted by focusing not only on what is visible on the image but also on the general context and common-sense knowledge. In this way, the resulting dialogues are more natural and complex. For example, in the Visual Dialogue task, the goal is to "imagine" the scene by asking questions about it based solely on a description of the image and the previous question/answer pairs. Therefore, answers tend to be concise and to the point. Conversely, in IGC the aim is to have a conversation on a topic centred on an image. Both parties have access to the image and the conversation can evolve naturally, perhaps without even explicitly mentioning the objects in the image (see Fig. 2.1 for an example of the differences between Visual Dialogue and IGC). To achieve this, Mostafazadeh et al. [26] collected a dataset of interactions elicited by social-media-like posts containing images.

Another area of active research is language grounding [15, 28, 40, 5, 4, 39, 25, 30, 32, 16], including grounded video descriptions [3, 50]. For example INGRESS [32] approaches the grounding of referring expressions in a robotic context to pick and place objects following natural language instructions. Besides handling "unconstrained" object categories and rich language expressions, the system can also interactively ask clarification questions to resolve ambiguities. The approach works in two stages; in the first stage, visual descriptions of objects are generated and compared against the referring expressions, and a list of candidate objects is built. In the second stage, pairwise relations between the candidate objects are assessed and the most likely object is chosen.

Parde et al. [28] ground language on visual concepts through a game-play strategy. Concretely, the system first learns concepts from images paired with natural language descriptions and then, it refines its knowledge by asking yes-no questions to a user in the context of a game of "I spy". By doing so, the system can adapt its concept representations to better align to that of its user. However, objects seen during game-play are not novel, i.e., they have been seen during the initial training phase. Thomason et al. [40] train their system in a similar fashion, i.e., through a game

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

of "I spy", but they also incorporate other modalities such as sound, haptic and proprioceptive. Moreover, both the user and the system take turns asking questions, with the other party tasked with guessing the referred object. In this case, the authors do evaluate generalisation by testing the system on "novel" objects. As opposed to the work by Parde et al. [28], a description of an object was given at the start of the turn and then the corresponding party proceeded to guess the object until the correct one was identified. In other words, no further questions about attributes were asked. During an initial stage, the system collected sensory data that it could then amass to further train its classifiers once the game was over. That is, in every subsequent game, the system plays with a new subset of objects but it already possesses features for all the objects. The training of the predicate classifiers though is updated after each game with the information collected in the previous games, i.e., which features describe which predicates. Building on the previous work, Thomason et al. [39] implemented a system for improving language parsing and grounding through human-robot dialogue in order to better deal with natural language instructions containing compositional language. The system uses bootstrapping from clarification dialogues to improve semantic parsing, and active learning to incorporate new concepts to its knowledge base. Object representations are built as before from data collected during an exploratory phase using visual, auditory, haptic and proprioceptive modalities [40].

2.1 Semantic Scene Graphs

Semantic scene graphs are data structures that encode the semantic information of a visual scene into a graph, where nodes represent instances of some sort, and edges represent their relationships. This allows for more compact and interpretable representations that can be leveraged for inference about the scene. Scene graphs have previously been used for object recognition [20], action recognition [1], image retrieval [17] and 3D scene generation/retrieval [13]. For example, Aksoy et al. [1] approached the categorisation of object-action relations by means of semantic scene graphs. Concretely, given a sequence of images of a scene, each image is first segmented and each region is treated as a node in a graph. Edges, representing relationships between segments, are then added depending on how those segments interact with each other. For instance, they consider four different spatial relations, i.e., *absence*, *no connection*, *overlapping*, and *touching*. The first relation simply indicates that the corresponding segment (entity) is absent from the scene whereas the second relation indicates that there are no connections between two segments. *Overlapping* indicates that one entity contains the other, and *touching* indicates that the segments partially overlap but neither is contained by the other. Changes in the topology of the graph are a sign of an event occurring in the scene, e.g., an object being moved. Actions then are represented as "event tables" which encode the different changes in the scene graph and similarities between actions can be computed from these event tables. Additionally, objects can be categorised based on the roles they play in the actions.

Johnson et al. [17] on the other hand used scene graphs for semantic image retrieval. This is done via a Conditional Random Field (CRF) model that reasons over possible groundings of test images and ranks them based on their likelihood. Critically, they don't address scene graph generation from natural language queries, which would be essential for real-world applications, particularly in our area of interest. The next section discusses this topic.

2.1.1 Scene Graph Generation

There is currently a large body of work concerned with the generation of scene graphs from images [22, 27, 43, 45, 21, 44, 48, 49, 23, 38, 47, 46, 18, 24, 37] as well as from point-clouds [7, 41, 42]. Xu et al. [43] proposed the generation of scene graphs via iterative message passing. As it is commonly done in the literature, Faster R-CNN [31] is used to identify bounding boxes of objects and contextual information is used to jointly infer object classes and their relationships. Context is propagated via message passing between two subgraphs, one processing node information and another processing edge information. By iterating this process, predictions can be refined and inference efficiency can be improved.

Zellers et al. [48] looked at the role of "motifs", i.e., regularly appearing subgraphs, for the automatic generation of semantic scene graphs. They observed that usually, object labels tend to be a good predictor of relationship labels but the opposite is not true. Based on this insight, they proposed a baseline that predicts relationship labels based on the frequency they appear in the training set between corresponding object labels. Furthermore, they proposed Stacked Motif Networks, a neural architecture that represents global context via bidirectional LSTM networks. Concretely, Faster R-CNN [31] is used to predict bounding boxes, over which global context is then computed. This context is then used by another LSTM network to predict the labels of the bounding boxes. Finally, another set of bidirectional LSTM networks is used to compute and propagate information for predicting edges given the bounding boxes, their labels and the global context computed so far.

Graph R-CNN [44] combines a network for proposing relations between nodes and an attentional mechanism for integrating contextual information into the resulting scene graph. Once objects have been detected via a Faster R-CNN

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

model [31], a Relation Proposal Network (RePN) is trained to compute a measure of relatedness between object pairs, allowing for the pruning of unlikely relations. Finally, an attentional Graph Convolutional Network (aGCN) is applied to the resulting sparsely connected scene graph to propagate contextual information across the graph by updating the representations of objects and relations based on their neighbours.

Zhang et al. [49] argue that most scene graph parsing approaches suffer from two major issues as a result of dividing the pipeline into two stages, i.e., an entity detection stage and a predicate prediction stage, and using a cross-entropy loss over predicate classes. The first issue relates to the confusion of different instances of the same object class in the scene graph. The second issue arises from the proximity between multiple subject-predicate-object triplets with the same predicate, which leads to ambiguity. To address these issues, a set of contrastive losses was proposed which aim to resolve these ambiguities.

Tang et al. [38] attempt to address the problem of bias in scene graph generation, i.e., the consequence of high imbalance in the distribution of relationships within labelled datasets. For example, they argue that relationships like "near" tend to dominate in comparison to "behind" or "in front of", and previous approaches tend to collapse richer relationships into more general ones, e.g., "person walks on/sits on/lays on beach" are collapsed into "person on beach". In order to address this issue, they proposed the use of causal inference. This method can be applied to any previous approach for removing biased predictions.

Suhail et al. [37] proposed an energy-based learning framework to address the issues of biased predictions based on the training data, and the lack of structure learning, e.g., by using a cross-entropy loss that ignores dependence between different objects and their relationships, in many prior approaches. The incorporation of the scene graph structure acts as an additional constraint which allows for more efficient learning using less data. As it was the case for the work by Tang et al. [38], this approach can be used in conjunction with other scene graph generation algorithms to leverage graph structure into the learning pipeline.

Despite the large volume of work on the automatic generation of scene graphs, much progress remains to be made. On the one hand, a large proportion of the detected relationships tend to be broadly general, e.g., cars have wheels and people have heads, and as such, they are not interesting from a dialogue point of view. On the other hand, the proportion of correct triplets (based on the ground-truth) found in the top K results is usually quite low. This does not necessarily mean that the models are failing but it underlines the difficulties of evaluating models attempting to automatically generate scene graphs.

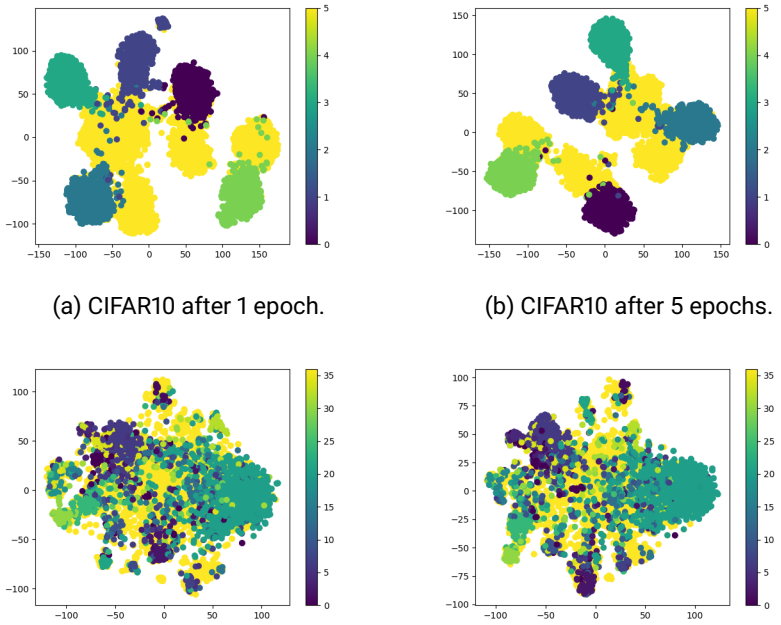
This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

3 Semantic Segmentation and Unknown Object Detection

Here we provide a short update on object detection and classification, compared to D2.4, which is based on semantic scene interpretation. This object detection is used to detect and build a semantic model of the scene, which is described in the next section.

Identifying unknown objects is crucial for the robot to understand the scene. The problem of classifying a new object as known or unknown is usually called Out-Of-Distribution Detection (OOD) or Open Set Recognition (OSR). While the OOD main focus is identifying instances which do not fit the known distribution, OSR focuses on both preserving closed set accuracy and unknown classification. Considering semantic or instance segmentation and object detection, the central common part we focus on is improving classification in terms of OSR. So our latest experiments are focused on Open Set Classification.

Recent works on open set recognition use distance to the cluster centre as a measure of anomaly score [8] or k-NN on a feature space [14]. These methods don't work well on features produced by the last layer of the closed set instance segmentation/object detection model. So the features from the right stage of the network need to be used, and the appropriate loss function needs to be selected to make features from the same known class close to each other and far from other known class features. We can use a simple idea of minimising the distance to the cluster centre inside this cluster from [8]. We used a subset of crops from the MS COCO dataset to test this approach. While for a simple dataset like CIFAR10 (figure 3.1), this method can separate the clusters well, the results on more complex and unbalanced data are worse. So further investigation needs to be done in this direction to improve our existing object detection and classification scheme used for semantic object detection and localization in 3D maps described in the next section.



(a) CIFAR10 after 1 epoch.

(b) CIFAR10 after 5 epochs.

(c) MS COCO crops after 1 epoch.

(d) MS COCO crops after 5 epochs.

Figure 3.1: Visualization of known and unknown (yellow) feature vectors used to measure anomaly score after non-linear dimensionality reduction using t-SNE.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

4 Object Detection and Localization in 3D Map

4.1 Current Segmentor/Detector

Our current approach is utilizing a meta-classifier, as proposed in [1], extending upon YOLACT [2], an object detection and instance segmentation network, whose output, given an image, is a vector of tuples (s, c, m, ...) containing information about the individual detections in the image, including the maximal softmax score s, corresponding class c and segmentation mask m. The meta-classifier accumulates these detections into clusters in 3D, which indicate potential presence of an object and tracks the distribution of YOLACT classification scores per each cluster (see fig 4.1) . Based on this distribution, a decision is made to either dismiss the possibility of an object being present, or updating its memory of 3D classifications, classifying it either as one of the 80 known classes present in the Microsoft COCO dataset [3] or acknowledging, that the class of the object is unknown. The memory is updated perpetually with each image, giving the possibility to reason retrospectively, even dismiss an already accepted detection. (See figure 4.2 for visual overview of this process)
The method for clustering detections is similar to that described in [1], assuming objects to be spheres of constant radius centered around a single point p. While rudimentary and by design introducing certain issues, more on that below, this method proved to be sufficient for initial testing. The current approach differs slightly from [1] in the way p is computed and detections are clustered, adjusting for missing depth data in images, and accounting for mask size when clustering spheres together.

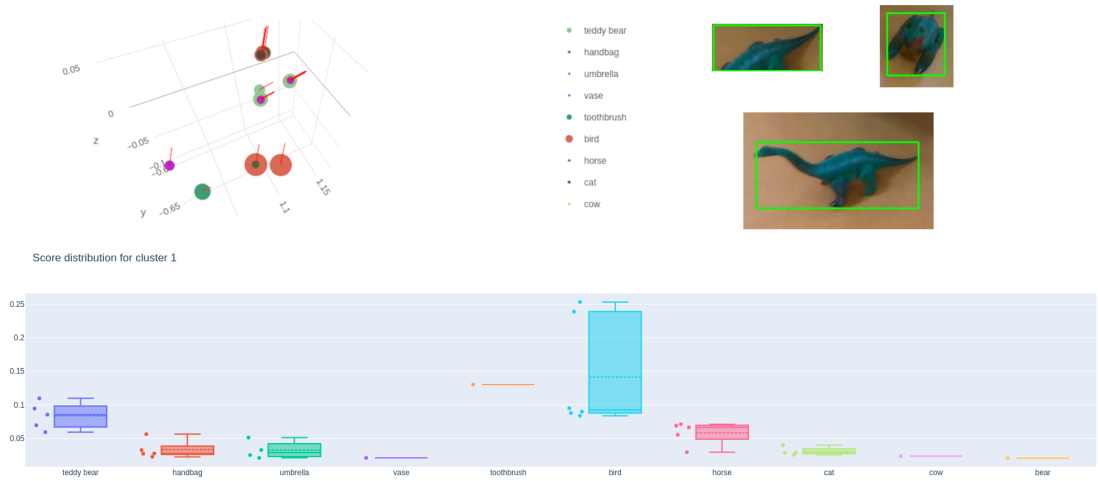


Figure 4.1: Visualization of a single cluster. On the top left is a plot of 3D back-projections of YOLACT detections that were clustered to this cluster, colored by class and scaled by confidence. The red lines point in the direction of the camera when detection occurs. Notice, how this direction seems to correlate with the favored class of the detection. On its right are images of some of the detections. Below is a box plot of detection score distribution.

4.2 Integration with ARI

The overall flow of the 3D semantic instance segmentation on ARI can be seen in figure 4.3. The system takes in as inputs the RGB image, its corresponding depth image, the camera intrinsic parameters (focal length, principal point), and the camera pose (rotation, translation) in the map frame of the map built from Visual SLAM.

The ROS [35] workflow is shown in figure 4.4. The tf library function $lookupTransform$ is used to obtain the

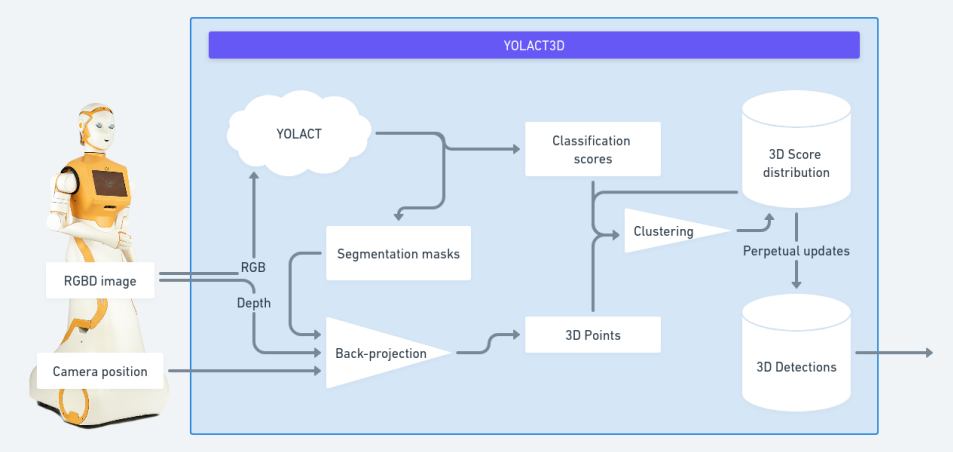


Figure 4.2: Overview of the 3D semantic instance segmentation process

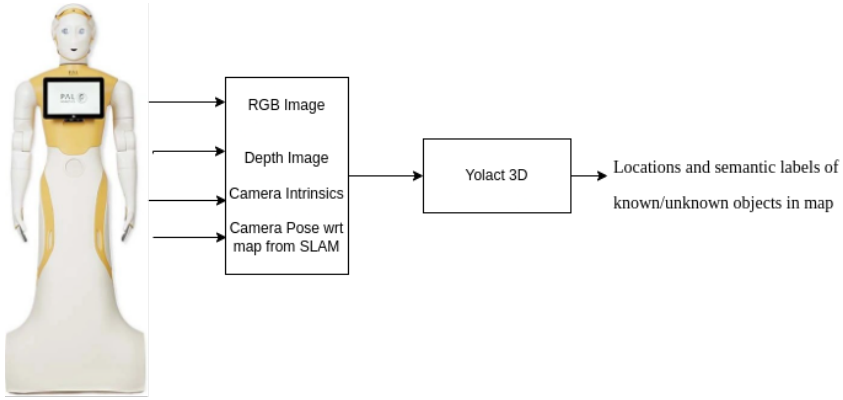


Figure 4.3: High level overview of flow of the 3D instance pipeline in ARI. The data is from the RealSense D435i torso front camera; The camera pose is computed from the robot pose with the tf library

transformation between the (source) map and the (target) camera frame $torso\_front\_camera\_color\_optical\_frame$. The output is published in the topic $/yolact3d/detected\_objects\_distributions$, which is a representation of detected objects with their probability distribution among possible classes (or unknown class), and their position in the map frame. This topic publishes messages of a custom ROS msg type of the form:

$$\text{Header header}$$

$$\text{object[] threeDobjects}$$

where $object$ is a msg of the form:

$$\text{Header header}$$

$$\text{string[] objects}$$

$$\text{float32[] probabilities}$$

$$\text{geometry\_msgs/Point position}$$

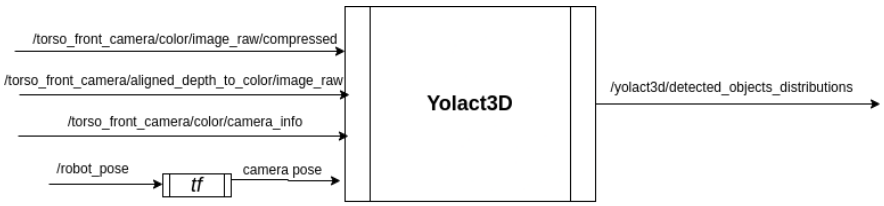


Figure 4.4: ROS workflow

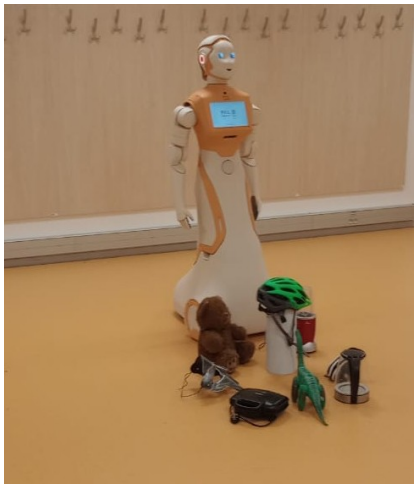


Figure 4.5: Data Collection where ARI was made to move autonomously in circular trajectories around the objects

4.3 Experiments

Here we describe first experiments with object detection and localization in 3D maps integrated in ARI.

4.3.1 Data Collection

Procedure

1. The room where the data is to be collected is mapped with ARI. Once complete, this map is loaded and ARI is set to localization mode.
2. Objects of varying sizes such as a skateboard, computer mouse, water bottle, coffee mug, figurines, etc., are placed on the floor.
3. ARI is manually controlled to move around and record relevant topics to a rosbag.

With this procedure, data collection was performed and the algorithm. However, the results were poor since the data was found to be of poor quality. The manual motion of ARI did not capture the objects sufficiently well, and the objects in 3D space could not be clustered as separate objects.

As a result, a modified procedure was followed for a new round of data collection.

2. Objects of varying sizes are placed around the floor near in a small radius around a point. See fig 4.5
3. ARI navigates autonomously in a trajectory of concentric circles, looking towards the center, of multiple radii around the objects. See fig 4.6

The circular trajectory ensures that ARI captures the objects from all orientation, and multiple radii makes ARI view the objects from varying distances. ARI moves to points along the circle, facing the center of the circle (where the objects are) at user specified intervals.

4.3.2 Topics Recorded

The following topics were recorded in rosbags:

1. front fisheye image (not used for current vision): /front_camera/fisheye/image_raw/compressed
2. Robot pose in map: /robot_pose
3. Tf: /tf and /tf_static
4. Torso front camera image: /torso_front_camera/aligned_depth_to_color/image_raw
5. Torso front camera parameters: /torso_front_camera/color/camera_info

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

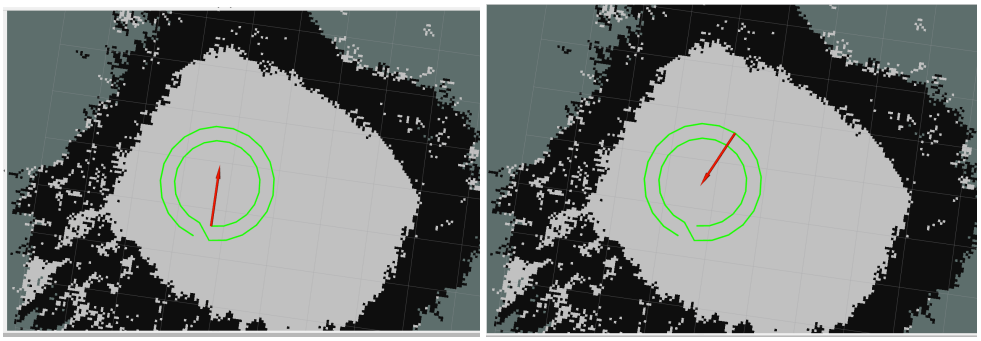


Figure 4.6: Circular trajectory around objects with goal poses along the circle looking at the objects. This ensures that ARI captures the objects at different angles, and viewpoints

4.4 Results

Overall, three sets of data was recorded (4.7), each observed in an interactive data visualization interface (example in figure 4.8) and evaluated, due to the small scale of the test, manually.

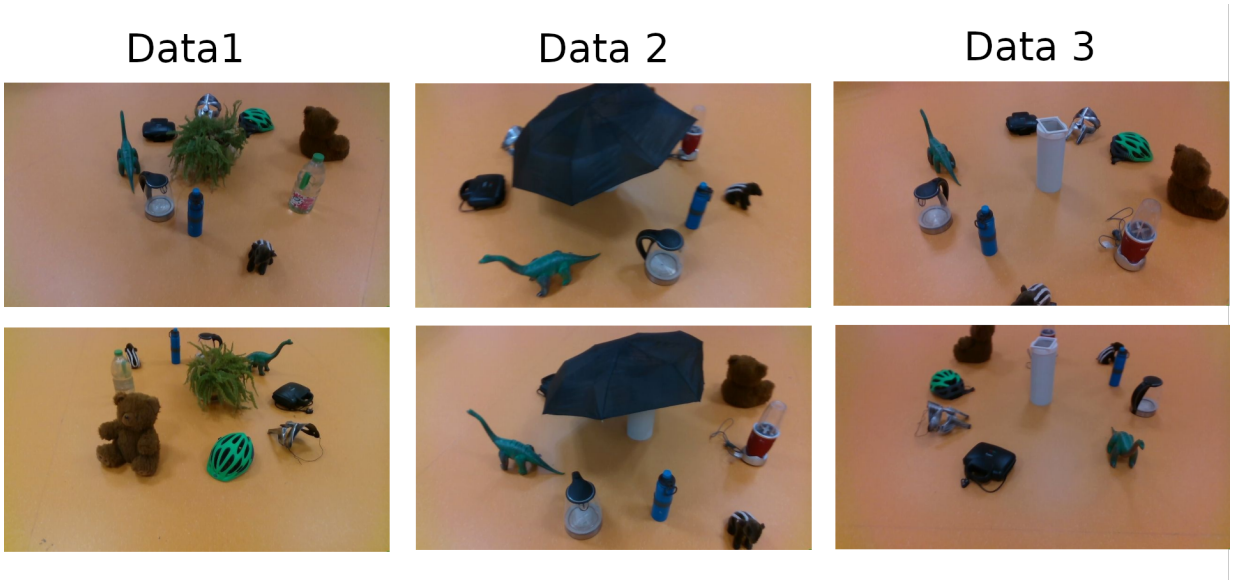


Figure 4.7: The three observed data recordings

Used metric for measuring performance of the model consisted of recall (eq:rec) and precision (eq:pr) at both differentiating unknown from known and detecting that an object is present. As every known class was detected and classified correctly, no metric was required for known class classification.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{4.1}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{4.2}$$

Results are presented in figure 4.9.

Probable cause of failure cases in object detection is the previously mentioned representation of objects as spheres of constant radius, which as consequence may result in one object having multiple clusters representing it (this is treated as false positive detection), or, having YOLACT detections of another object clustered to the cluster which represents it. The first case was especially notable notable in Data 2 as it contained an open umbrella, which spans a relatively large area, the second case was rare enough to not cause significant interference in most classification decisions, however there are two cases, where an object was not detected as its YOLACT detections were clustered to clusters representing other objects.

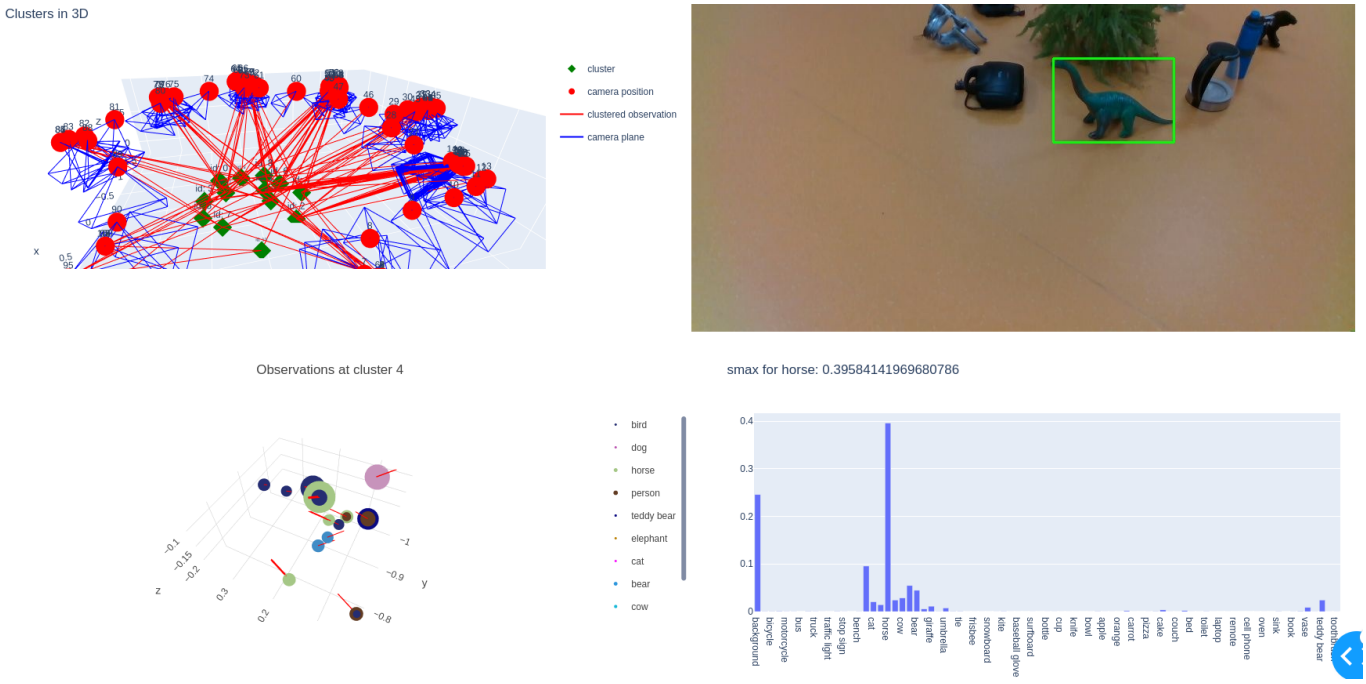


Figure 4.8: The interface used for evaluation. Top left contains an interactive 3D visualization of camera and cluster positions. Below is a visualization of detections in a specific cluster from figure 1. Top right contains an image with a specific detections' bounding box. Below is YOLACT softmax score for that detection.

Data 1		
True \ Predicted	**Known**	**Unknown**
Known	4	2
Unknown	0	4

	Detection	**Unknown**
Recall	1	0.66
Precision	0.77	1

Data 2		
True \ Predicted	**Known**	**Unknown**
Known	3	0
Unknown	0	7

	Detection	**Unknown**
Recall	0,9	1
Precision	0.66	1

Data 3		
True \ Predicted	**Known**	**Unknown**
Known	2	1
Unknown	0	5

	Detection	**Unknown**
Recall	0.89	0.83
Precision	0.89	1

Figure 4.9: Results of the experiment. Above is the confusion matrix for Known vs Unknown classification. Below are the precision and recall scores for object detection (not classification) and classification of Unknown among Known (Unknown is regarded as positive).

Despite these addressable errors, it seems that our method is a good starting point for the task of open-world segmentation / detection and stands as a proof of concept for a meta-classification approach.

5 Semantic Scene Graphs for Visual Grounded Dialogue

It is crucial for social robots in HRI to be able to hold coherent conversations in visual/spatial situations where objects and their relations are critical for task success (e.g. directions to a place or object). For SPRING we explore to what extent we can use more visually grounded semantic representations of utterances within the multimodal dialogue context developed in the system. Our video demonstration at HRI 2021[1] [29], was a first step in this direction as it uses deep learning for object recognition to create spatial representations which are used to answer questions from the user.

The goal of this module is to leverage semantic scene graphs to guide visually grounded conversations (the proposed system architecture is illustrated in Fig. 5.1). Concretely, by having access to a graph representation of the visual scene, the system should be able to reason about the location of objects in order to answer user queries as well as to produce meaningful descriptions. Moreover, by combining such abilities with task-based and open dialogue, the system should be able to engage in richer conversations with its users. However, there are many challenges to overcome when using scene graphs. When automatically generating them, the results tend to be extremely noisy with many misclassifications and the estimation of incorrect relationships between objects. Furthermore, the datasets on which the scene graph generation models are trained are usually biased and incomplete, i.e., some relationships are more predominant than others and not every possible relation is specified. In particular, many of the indicated relationships tend to refer to the description of structural relationships which are generally known to be true, e.g., cars have wheels and people have heads, and thus may not be informative from a dialogue perspective. Other relationships may only be interesting based on the context, e.g., the fact that a person is wearing a shirt is not interesting unless we are looking for a person wearing a particular type of shirt. Likewise, a car missing a wheel is more interesting than the fact that cars usually have four wheels.

Another related problem stems from the fact that multiple relationships exist for the same pair of objects and determining which one, if any, is the correct one is not trivial. Sometimes, relationships with high confidence are wrong whereas relationships with low confidence can be right. This also applies to the results of object detection and classification. In some cases, interesting objects can have a low confidence score and in many cases, there are multiple detection results for the same object.

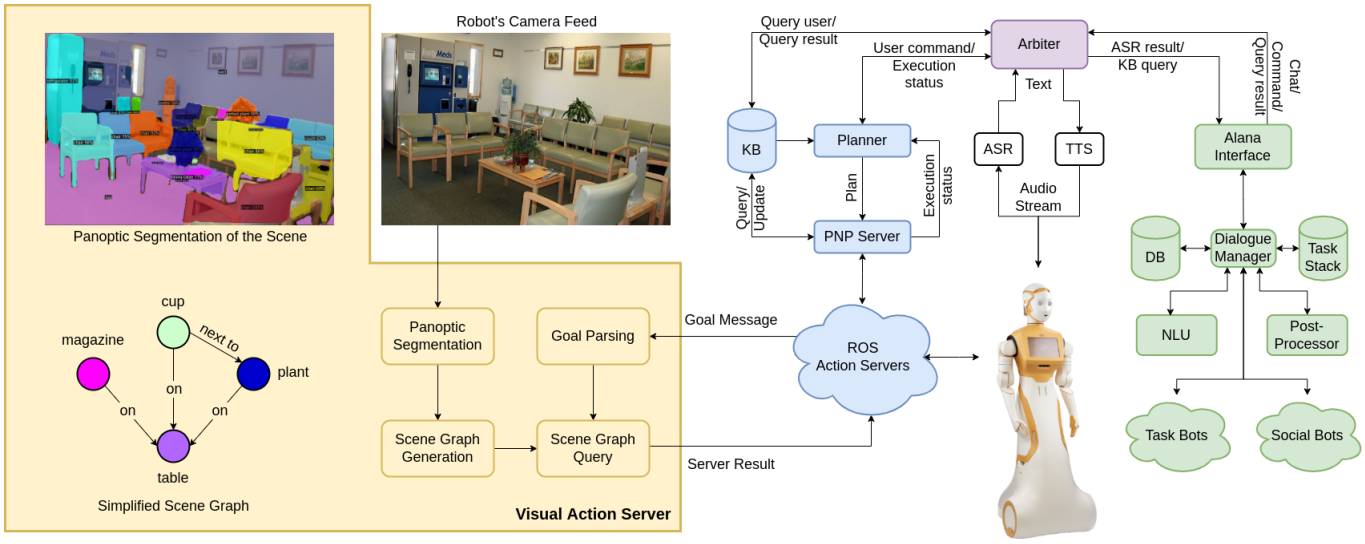


Figure 5.1: Proposed system architecture for visually grounded situated dialogue.

[1] https://www.youtube.com/watch?v=eY_BNxwrlPg

5.1 Scene Graph Generation

The proposed approach generates scene graphs by an end-to-end scene graph generation model Relation Transformer (RelTR) [10]. RelTR (Fig. 5.2) is a one-stage end-to-end framework for scene graph generation based on the Transformer encoder-decoder architecture. The encoder reasons about the visual feature context while the decoder infers a fixed-size set of triplets <subject-predicate-object> using different types of attention mechanisms with coupled subject and object queries.

Compared with other methods, RelTR achieves state-of-the-art performance using only visual appearance, with very few model parameters and fast inference. RelTR is a one-stage method that predicts sparse scene graphs directly only using visual appearance without combining entities and labeling all possible predicates, in contrast to the previous scene graph generation methods, presented in section 2.1.

The model was trained on Visual Genome [19], a large dataset of images annotated with objects, their attributes, and relationships to other objects, natural language descriptions, and question/answer pairs. In order to prune the scene graph, we currently set confidence thresholds for both object detections and relationships. Additionally, we keep a list of "relevant categories" that is used to filter the scene graph prior to pruning. RelTR can predict the subjects (blue), objects (orange) and their predicates simultaneously with learned subject and object queries (Fig. 5.3).

Currently, the scene graph generation module is implemented as a ROS action server. Given an empty goal, the server processes the current image from the robot's camera feed and returns the scene graph information by means of a custom message that contains fields for the object bounding boxes (*bbox*), labels (*bbox_labels*) and scores (*bbox_scores*), and for the relationship pairs (*rel_pairs*), their labels (*rel_labels*) and scores (*rel_scores*). By interpreting this information, an action client can produce dialogue responses that leverage the semantic information available in the scene graph. Fig. 5.4 shows an example of an action client interacting with the scene graph server. The relationship pairs contain the indices of the interacting objects as they appear in the corresponding data structures, i.e., *bbox*, *bbox_labels* and *bbox_scores*. For example, if the first relationship pair (index 0 in *rel_pairs*) is (1,3), that means that the objects in the second (index 1) and fourth (index 3) positions interact through the first relationship listed in *rel_labels*.

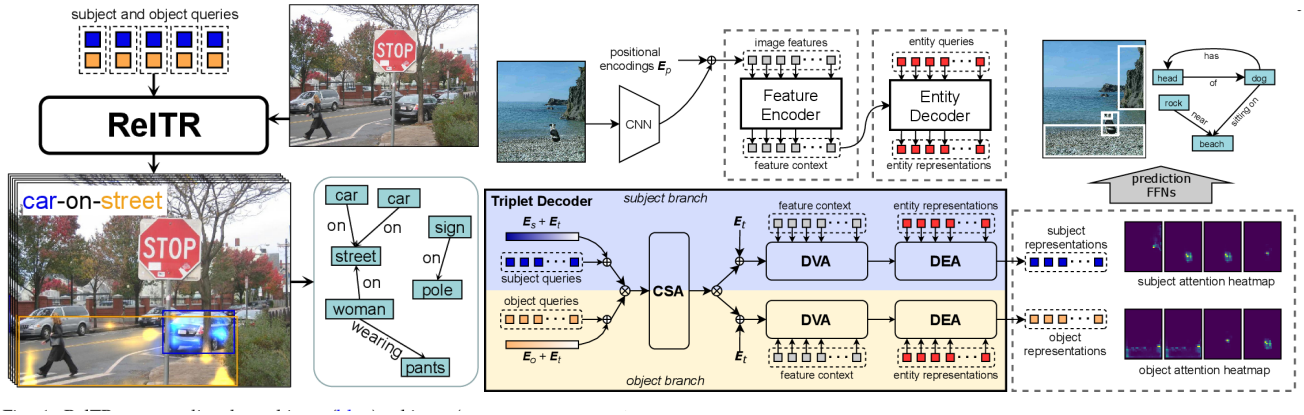


Figure 5.2: RelTR encoder-decoder architecture [10]. A pair of subject and object representations with attention heat maps is decoded into a triplet <subject-predicate-object> by feed forward networks (FFNs). CSA, DVA and DEA stand for Coupled Self-Attention, Decoupled Visual Attention, and Decoupled Entity Attention. Ep, Et, Es, and Eo are the positional, triplet, subject, and object encodings respectively.

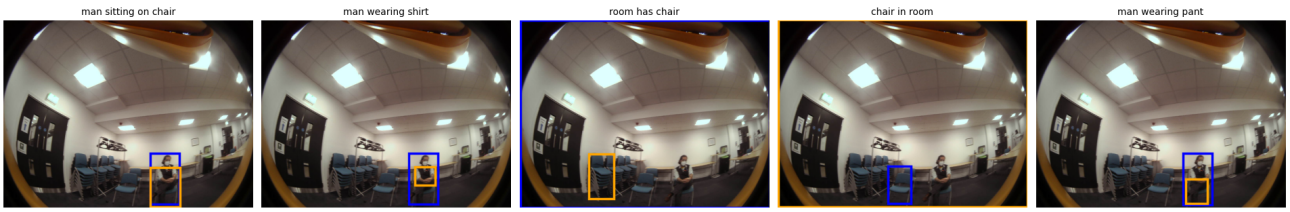


Figure 5.3: Visual Dialogue examples with RelTR and ARI robot cameras in a mock-up hospital reception waiting room at HWU.

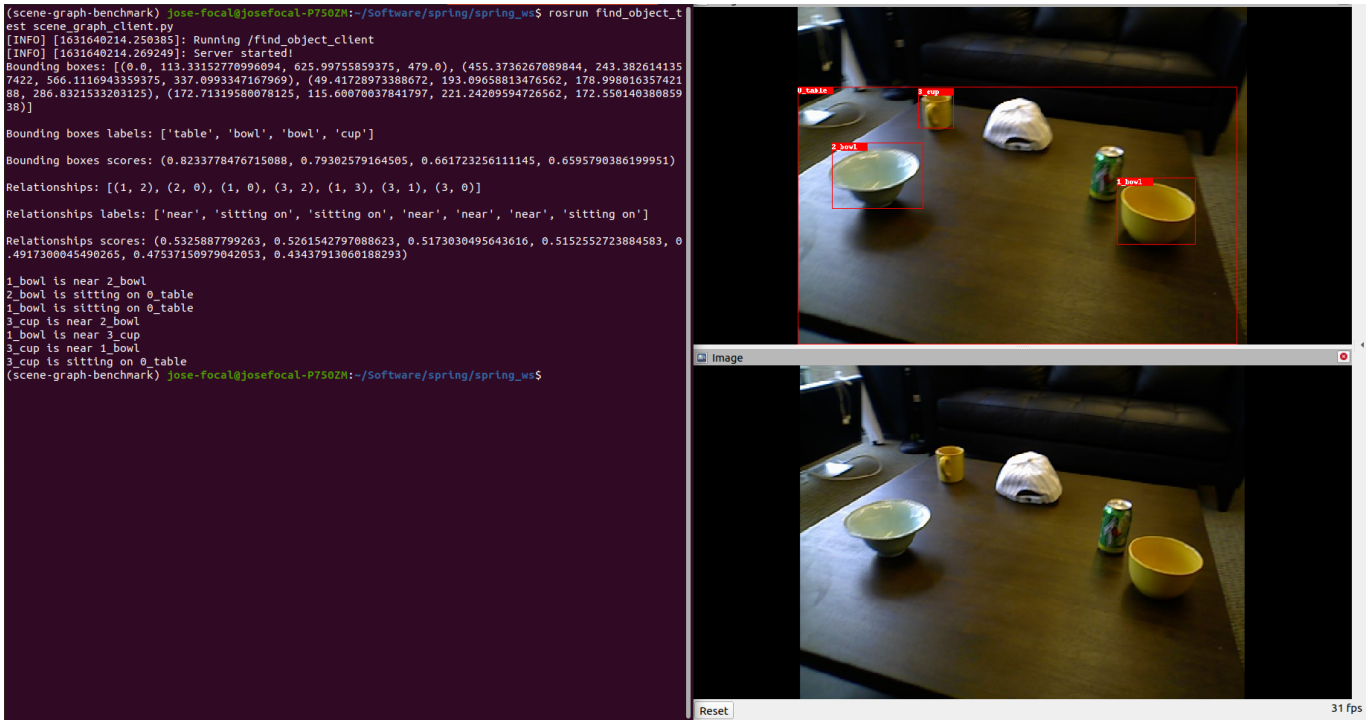


Figure 5.4: Example of action client parsing the information available in the scene graph.

5.2 Response Generation

As depicted in Fig. 5.1, the ROS Action Servers interact with the PNP Server which sends requests to the former and relies the responses to the dialogue system through the Arbiter. An earlier version of the system relied on manually created scene graphs. An action server would then receive an image ID, retrieve the corresponding scene graph and generate a response based on the object of interest specified by the user, and its location according to the scene graph. The current version instead consists of a response generation action server that communicates with the PNP server and also acts as a client to the scene graph action server (see Fig. 5.5). In this way, whenever the response generation action server receives a request from the PNP server, it calls the scene graph action server which provides the graph of the current scene. The response generation is then carried out in the same way it was for the manually created scene graphs.

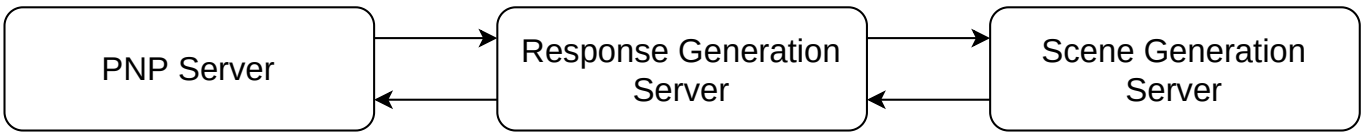


Figure 5.5: Communication between the different servers involved in extracting the scene graph and generating a response from it.

5.2.1 Identified Issues

The way the system's response is currently being generated, i.e., via rules, has many limitations. First of all, it aggregates relationships concerning the target object in a single sentence assuming that the corresponding predicates describe spatial relationships. However, predicates can describe richer relationships, e.g., possession, mereological properties, state, etc. For instance, consider an example of a scene graph (corresponding to the scene depicted in Fig. 5.6) represented by the following triplets (*object, predicate, object*):

(0_table has 3_handle)

(1_cup has 3_handle)

(1_cup near 2_bowl)

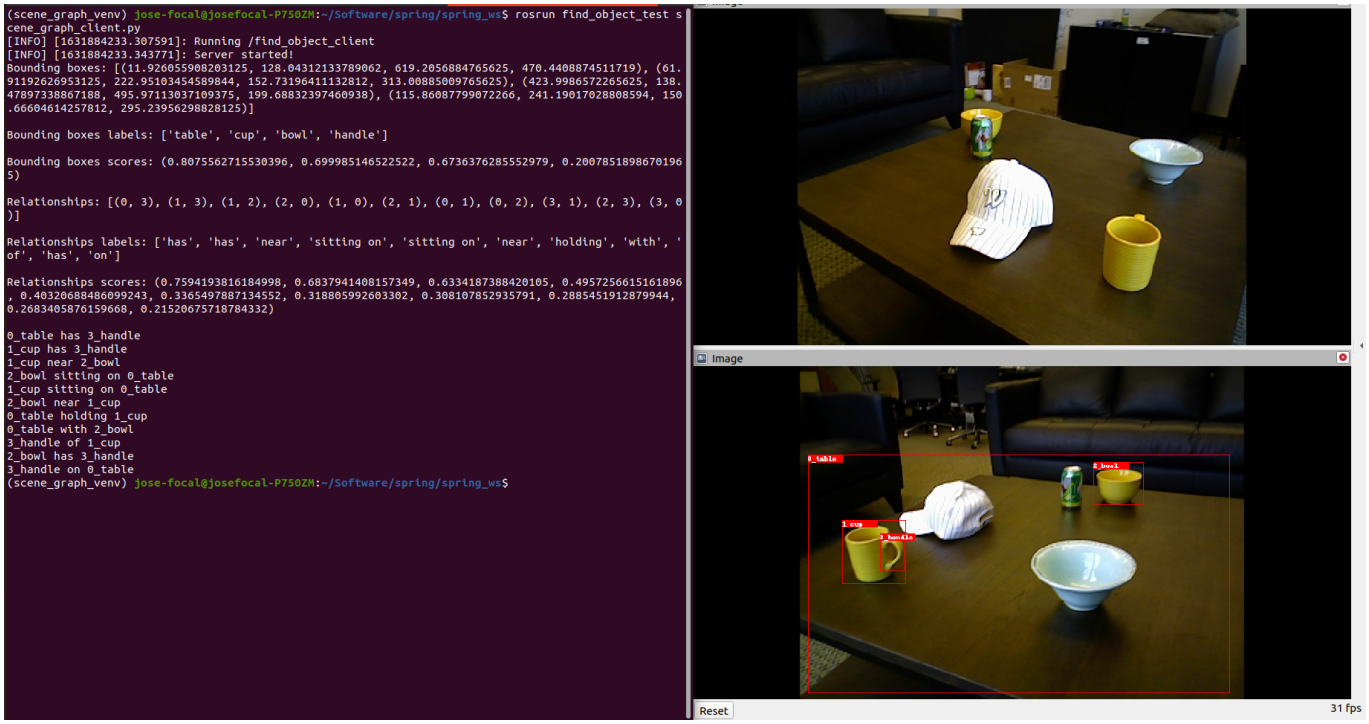


Figure 5.6: Example of scene depicting non-spatial relationships.

(2_bowl sitting on 0_table)

(1_cup sitting on 0_table)

(2_bowl near 1_cup)

(0_table holding 1_cup)

(0_table with 2_bowl)

(3_handle of 1_cup)

(2_bowl has 3_handle)

(3_handle on 0_table)

The snippet of code that produces the output utterance is given below:

```
relations = [item for item in scene_graph if item[0] == goal.object_frame]
if not relations:
    self._result.located = False
else:
    self._result.located = True
    self._result.quantity = 1
    for idx, rel in enumerate(relations):
        if not idx:
            self._result.spatial_relation = rel[2]
            self._result.location = "the {}".format(rel[1])
        else:
            self._result.location += " {} the {}".format(rel[2], rel[1])
```

where *relations* is simply the list of triplets given above, *goal.object_frame* represents the object referred to by the user, and *self._result* stores the first relationship for the target object and its location, which is the result of aggregating every other available relationship for the target object. For example, assuming we ignore objects' IDs (i.e., the preceding numbers in the object names), if the user asked for the *cup*, then *self._result.spatial_relation* would be *has*, and *self._result.location* would be *the handle near the bowl sitting on the table*. Then, the dialogue system, would produce a response like "There is a cup *has* the handle near the bowl sitting on the table." because it assumes that *self._result.spatial_relation* denotes a spatial relationship and *self._result.location* is an actual location. Of course, this could be addressed by filtering relationships based on their type but this would merely be putting a patch over the issue. Additionally, in the example shown above, we can see that the scene graphs tend to be corrupted with wrong

relationships, e.g., *(0_table has 3_handle)*, which highlights the limitations of current automatic scene generation algorithms.

The second issue relates to the fact that we are currently not distinguishing among objects of the same type. For instance, Fig. 5.4 shows a scene with two bowls and a cup placed on a table. The retrieved graph for this scene according to the used model is:

(1_bowl near 2_bowl)

(2_bowl sitting on 0_table)

(1_bowl sitting on 0_table)

(3_cup near 2_bowl)

(1_bowl near 3_cup)

(3_cup near 1_bowl)

(3_cup sitting on 0_table)

If the user asks for the *cup*, then there will be two relationships (*cup, near, bowl*) and the resulting output utterance will be something like "There is a cup near the bowl near the bowl on the table.". In order to address this issue for now, we remove duplicate relationships (without accounting for the object's IDs) although this does not include symmetrical relationships. More elegant solutions could involve distinguishing between the different objects by using attributes. However, it has to be noted that depending on the context, some relationships may not be relevant and should be avoided when producing the output utterance. Likewise, in the previous example it is unnecessary to refer to both bowls when describing the location of the cup even if we were able to distinguish between them. In other words, saying "There is a cup near the blue bowl on the table." is preferred to saying "There is a cup near the blue bowl near the yellow bowl on the table.".

5.2.2 Scene Graph Pruning

As illustrated above, there are several areas for improvement by future work is the response generation. Once the scene graph has been relayed to the dialogue system, it is necessary to select the most relevant triplet based on dialogue context and generate a coherent Natural Language response.

In future work, such tasks could be treated as a conditional generation problem for Vision-and-Language models such as VL-T5 [9], given suitable data for fine-tuning.

Rule-Based Pruning

As discussed earlier, we currently set confidence thresholds to prune the scene graph. This could be complemented with a trained model or additional heuristics which should allow to reduce the thresholds values without significantly corrupting the graph. An example of such heuristics could be the identification of multiple detection results corresponding to the same object and keeping the one with the highest classification score or with the largest bounding box. In this case, highly overlapping bounding boxes with the same associated label would be merged such that a single result is kept. This involves also merging all the associated data, e.g., relationships to other objects.

Binary Classification of Triplets

Another potential way to address the issues identified before is to train a binary classifier to estimate which triplets may be relevant and which may not. Such a strategy requires the creation of a dataset based on the scene graph results given by the chosen model. Alternatively, an existing dataset (e.g., the one used to train the model to detect scene graphs) could be used where the ground-truth provides the positive examples and every other triplet output by the model is set as a negative example.

This project has received funding from the European Union's Horizon 2020
Research and Innovation Programme under Grant Agreement No. 871245.

6 Conclusions

The software modules describe in this deliverable will be made available on the SPRING project Gitlab repositories for Work Package 2 [33], object detection and localization in 3D map based on semantic segmentation and unknown object detection (Chapters 3 and 4), and Work Package 5 [34], semantic scene graphs for visual grounded dialogue (Chapter 5). The software packages use ROS (Robotics Operating System) [35] to communicate with each other and with the modules developed in the other work packages.

These will be available to the public for the duration specified in the SPRING project proposal. As per European Commission requirements, the repository will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software from the project coordinator at `spring-coord@inria.fr`.

Next we plan to connect object detection and localization in 3D map based on semantic segmentation with semantic scene graphs for visual grounded dialogue to be able to detect unknown objects and learn their labels by inquiring users. This open new challenges in sharing the knowledge of represented in a robot with the knowledge of human users.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Bibliography

[1] Eren Erdal Aksoy, Alexey Abramov, Florentin Wörgötter, and Babette Dellen. Categorizing Object-Action Relations from Semantic Scene Graphs. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 398–405, 2010.

[2] Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K. Marks, Chiori Hori, Peter Anderson, Stefan Lee, and Devi Parikh. Audio-Visual Scene-Aware Dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7558–7567, 2019.

[3] Muhannad Alomari, Eris Chinellato, Yiannis Gatsoulis, David C Hogg, and Anthony G Cohn. Unsupervised Grounding of Textual Descriptions of Object Features and Actions in Video. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning*, pages 505–508, 2016.

[4] Muhannad Alomari, Paul Duckworth, Nils Bore, Majd Hawasly, David C Hogg, and Anthony G Cohn. Grounding of Human Environments and Activities for Autonomous Robots. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[5] Muhannad Alomari, Paul Duckworth, David C Hogg, and Anthony G Cohn. Natural Language Acquisition and Grounding for Embodied Robotic Systems. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 4349–4356, 2017.

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015.

[7] Iro Armeni, Zhi Yang He, Jun Young Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5664–5673, 2019.

[8] Feiyang Cai, Zhenkai Zhang, Jie Liu, and Xenofon Koutsoukos. Open set recognition using vision transformer with an additional detection head. *arXiv preprint arXiv:2203.08441*, 2022.

[9] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation, 2021.

[10] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *ArXiv*, abs/2201.11460, 2022.

[11] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Stefan Lee, Jose M.F. Moura, Devi Parikh, and Dhruv Batra. Visual Dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–335, 2019.

[12] Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. GuessWhat?! Visual Object Discovery through Multi-Modal Dialogue. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4466–4475, 2017.

[13] Matthew Fisher, Manolis Savva, and Pat Hanrahan. Characterizing Structural Relationships in Scenes using Graph Kernels. *ACM Transactions on Graphics*, 30(4), 2011.

[14] Silvio Galesso, Max Argus, and Thomas Brox. Far away in the deep space: Nearest-neighbor-based dense out-of-distribution detection. *arXiv preprint arXiv:2211.06660*, 2022.

[15] Sergio Guadarrama, Lorenzo Riano, Dave Golland, Daniel Gouhring, Yangqing Jia, Dan Klein, Pieter Abbeel, and Trevor Darrell. Grounding Spatial Relations for Human-Robot Interaction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1640–1647, 2013.

[16] Felix Hill, Olivier Tieleman, Tamara von Glehn, Nathaniel Wong, Hamza Merzic, and Stephen Clark. Grounded Language Learning Fast and Slow. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[17] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Image Retrieval using Scene Graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.

[18] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W. Taylor, Aaron Courville, and Eugene Belilovsky. Graph Density-Aware Losses for Novel Compositions in Scene Graph Generation. In *Proceedings of the 31st British Machine Vision Conference (BMVC)*, 2020.

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123:32–73, 2017.

[20] Wen Jing Li and Tong Lee. Object Recognition by Sub-Scene Graph Matching. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1459–1464, 2000.

[21] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable Net: An Efficient Subgraph-Based Framework for Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 346–363, 2018.

[22] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene Graph Generation from Objects, Phrases and Region Captions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1270–1279, 2017.

[23] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. GPS-Net: Graph Property Sensing Network for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3743–3752, 2020.

[24] Hengyue Liu, Ning Yan, Masood S. Mortazavi, and Bir Bhanu. Fully Convolutional Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11546–11556, 2021.

[25] Jinpeng Mi, Jianzhi Lyu, Song Tang, Qingdu Li, and Jianwei Zhang. Interactive Natural Language Grounding via Referring Expression Comprehension and Scene Graph Parsing. *Frontiers in Neurorobotics*, 14, 2020.

[26] Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P Spithourakis, and Lucy Vanderwende. Image-Grounded Conversations: Multimodal Context for Natural Question and Response Generation. In *Proceedings of the 8th International Conference on Natural Language Processing*, pages 462–472, 2017.

[27] Alejandro Newell and Jia Deng. Pixels to Graphs by Associative Embedding. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 2172–2181, 2017.

[28] Natalie Parde, Adam Hair, Michalis Papakostas, Konstantinos Tsiakas, Maria Dagioglou, Vangelis Karkaletsis, and Rodney D. Nielsen. Grounding the Meaning of Words through Vision and Interactive Gameplay. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1895–1901, 2015.

[29] Jose Part, Daniel Hernandez-Garcia, Yanchao Yu, Nancie Gunson, Christian Dondrup, and Oliver Lemon. Towards visual dialogue for human-robot interaction. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021.

[30] Mihir Prabhudesai, Hsiao Yu Fish Tung, Syed Ashar Javed, Maximilian Sieb, Adam W. Harley, and Katerina Fragkiadaki. Embodied Language Grounding with 3D Visual Feature Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2217–2226, 2020.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[32] Mohit Shridhar, Dixant Mittal, and David Hsu. INGRESS: Interactive Visual Grounding of Referring Expressions. *The International Journal of Robotics Research*, 39(2-3):217–232, 2020.

[33] SPRING Project. Wp2: Environment mapping, self-localisation and simulation. `https://gitlab.inria.fr/spring/wp2_mapping_localization`.

[34] SPRING Project. Wp5: Spoken conversations. `https://gitlab.inria.fr/spring/wp5_spoken_conversations`.

[35] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. `https://www.ros.org`.

[36] Alessandro Suglia, Ioannis Konstas, Andrea Vanzo, Emanuele Bastianelli, Desmond Elliott, Stella Frank, and Oliver Lemon. CompGuessWhat?!: A Multi-task Evaluation Framework for Grounded Language Learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7625–7641. Association for Computational Linguistics, 2020.

[37] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-Based Learning for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13936–13945, 2021.

[38] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased Scene Graph Generation from Biased Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3713–3722, 2020.

[39] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Nick Walker, Yuqian Jiang, Harel Yedidsion, Justin Hart, Peter Stone, and Raymond J. Mooney. Improving Grounded Natural Language Understanding through Human-Robot Dialog. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6934–6941, 2019.

[40] Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J Mooney. Learning Multi-Modal Grounded Linguistic Semantics by Playing "I Spy". In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3477–3483, 2016.

[41] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3960–3969, 2020.

[42] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. SceneGraphFusion : Incremental 3D Scene Graph Prediction from RGB-D Sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7515–7525, 2021.

[43] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5419, 2017.

[44] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

[45] Michael Ying Yang, Wentong Liao, Hanno Ackermann, and Bodo Rosenhahn. On Support Relations and Semantic Scene Graphs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 131:15–25, 2017.

[46] Alireza Zareian, Svebor Karaman, and Shih Fu Chang. Bridging Knowledge Graphs to Generate Scene Graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–623, 2020.

[47] Alireza Zareian, Zhecan Wang, Haoxuan You, and Shih Fu Chang. Learning Visual Commonsense for Robust Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 642–657, 2020.

[48] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural Motifs: Scene Graph Parsing with Global Context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5831–5840, 2018.

[49] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical Contrastive Losses for Scene Graph Parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11535–11543, 2019.

[50] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. Grounded Video Description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6571–6580, 2019.