Chapter 1

Introduction

A common problem in visual question answering is the over-weighting of linguistic over visual features in the answering. [Fuk+16]. Research has shown that models can superficially perform tasks without learning the underlying reasoning process [**turmsimple**]. In visual question answering, it has been observed that a system cheats its way into answering the questions without taking the reasoning steps that humans would logically take to answer a question. [ABP16],[Zha+16], [Fuk+16].In particular,

In Such cases the system answers correctly by exploiting linguistic biases in the dataset, as it tends to rely primarily on the language model and ignore the visual information [Goy+17]

model learns biases in training and manages to give good results in the testing [Sel+20]. The underlying issue here is that the model answers by memorising prior textual information. For example, a neural network might answer the question "What covers the ground?" correctly by answering "snow", "not because it understands the scene but because biased datasets often ask questions about the ground when it is snow-covered." [Joh+17]; This learning problem is crucial because it makes it challenging to evaluate the model's improvements[Agr+18].

Learning to depend on the language model and disregard images indicates a "lack of sufficient visual grounding" [Agr+18]. Grounding is the process of connecting a symbolic representations such as symbol tokens (word) to a non-symbolic representation such as the sensory (visual) features of an object [Har90]. This connection would produce a "symbolic representation" grounded in the physical world. To put it in a simplified example, symbolic representation could be the picture of horse when we read the word "horse".

The concept of grounding and the cognitive theory behind it will be discussed more in a further section. The point that we want to make of this brief description is that "The robot must be able to refer to and resolve references to the environment"[Dob09]. Otherwise, lack of visual grounding means that the system does not comprehend what the words in the question refer to in the image.

Researchers improving VQA systems argue that a robot's capability to predict an answer based on a reasoning process would "require models to perform inference at multiple levels of abstraction" [Sel+20]. for example, "is the banana ripe?" where it would instantly answer "no". To answer this question, it would require the system to rely on perception to answer subquestions such as where is the object? What are its shape, size, and color? Then reason that the "yellow" color indicates ripeness.

1.1 Probelm formulation

In an experiment we conducted on the VQA model, we noticed that the system tends to answer questions relying mainly on the questions (bias). The question of the experiment is if the answer-prediction would change or still be correct if we ask the system about the color of the table in the living room and input a picture of a bathroom without a table in it. The results showed an increase in the total accuracy of the prediction despite the absence of the required visual information.

Yasmeen expirement ....

These results could indicate different things on the model and its data, but the least it could tell is that the system did not need to rely on the images to answer the questions correctly.

The observation of the results opens question-marks on two major components. The first is on the VQA model, as by why it tends to neglect the visual information and rely mostly on linguistic features. The second is a question on the dataset and its contribution to answering the questions correctly.Nonetheless, if one assumes that the model's ineffectiveness stems from a model-data mismatch, how would the system perform if asked different types of questions than the existing ones. This speculation becomes more relevant given that evaluated questions consist predominantly of one question type, color.

color questions is a topic itself, a list of sources [Mon+17]

1.2 Focus and research questions

We have an evidence that the dataset is simplistic, and contains bias in color question.

. Can the system answer more complex questions. (A usful robot should answer a variety of questions.)

. To what extent does the system use the visual information.

. How would the navigation model preform with new questions.

. Does asking more questions improve the system's reliance on vision in color question.

. Does the inclusion of spatial questions improve the system's learning of computational answers- such as olive-green, dark -blue.

. Would a tranformer-based based attention model improve the the preformance of the vqa model.

Adding new questions could help test the system's capabilities, but more importantly, we consider it a step to enhance the system's cognition.

The VQA system that we are improving is part of a robotic system that should ideally be helpful for human use. Social robot's usability is very dependent on its exhibition of human intelligence [FND03]. Hence that correct question-answer prediction does not necessarily indicate the system's ability to reason.

An example from the data presented in the Habitat project requires even fewer abstractions, "what color is the sofa?"; The system would only need to rely on perception answer itself "where is the object," then answer the color question.

However, color questions could get more complex as "people employ compositional color descriptions to express meanings not covered by basic terms, such as greenish-blue" [MGP16]. It would be shallow to assume that color questions are simplistic, especially if we expect the system to answer colors beyond the basic color terms like "green" and "red."

1.3 Related work

1.3.1 meaning

(Add subsection for meaning in the physical space, and how language and physical space influence each other)

The meaning of words is not a mere psychological phenomena. Our understanding of the physical world, manifested in language, stem from faculties such as perception and memory[Reg96]p,27. Perception, in particular, is central to our physical experience[Bar+99]. We conceive the physical world through perception; and we represent our conceptualization of the perceptual experience in words[LJ08],p59. Therefore, the meaning of a word, is not only bound up with linguistic characters and mental notion but also with some physical representation in the world. For example, the meaning of the word "chair" is represented by its token-characters (c,h,a,i,r), contains a perceptual symbolism(mental understanding), and refers to an entity with physical features in the world.

s robot do. 1.1 Language is Embedded in t

Distributional representation is a widely used method to represent meanings in words [Mik+13]. Word-meaning in these approaches is defined by its context–The meaning of a word is represented by the word and the words surrounding it. Using language to define language is successful, for example, to make inference that the "university" and "student" are close to each other given their common context of "education".This inferential ability is good for many tasks.

In the field of semantics, researchers seek to represent word-meaning by our mental and perceptual experience of the world. In particular,interactive tasks requires the exhibition of more intelligent behaviour; and this drives the necessity to have more meaningful representations by means of connecting 'words' to the physical word[Nil07]. To get rich meanings by connecting to the physical world would be to connect language with perception[Moo08].

1.3.2 Grounding meaning

In cognitive semantics, connecting language and perception means connecting low-level perceptual data with high-level meaning(language). low-level data is the representation of perceptional meaning, such as the sensory information from an image. "high-level" is logical inferential.

The ability to connect high with low level representation is important for any task requiring "seeing" and attending answer. "Symbol system problem"[Har90] is when a computational system process a textual and visual input and does not understand the perceptual reference of the text in the image. Grounding text in vision is when we connect the "high-level" symbolic representations such as symbol tokens (word) to a "low-level" non-symbolic representation such as the sensory (visual) features.

Research uses probabilistic models to connect the two domains. Traditionally, the probabilistic learning aims at draw an alignment between sentences, phrases, and words with the corresponding perceptual representations.[Low99].

There are three main approaches to probabilistic combination of perception language. The first is by finding the probability of a grammatical entity(text) being related to a perceptual representation. The second is by classifying each word in a sentence through probability distribution of words over a perceptual representation. The third is classification of word-embedding in a perceptual space.

[Lar17] Categorizes the three methods by their approach to meaning-representations:

1. Meaning as sets 2. Meaning as transparent function 3. Meaning as opaque function

Meaning as sets refers to the methods that use ......

In [Mat+12] [Lar15] we see examples of connecting formal semantics with perception.

[Lar17] evaluates the different methods in respect to compostionality in language. Composotionality is the notion that the whole meaning of a sentence compose of the the independent meaning given by its units (words). Four of the basis of the evaluation are the following:

1. Dealing with intersective compostionality: intersective compositionality is when two words which one is attributed to the meaning of the other "brown bear" where it means a bear that is brown-[brown and bear]

2. Dealing with non-intersective compostionality: non-intersective is one word does not modify the second, such as [Teddy bear]. 'Teddy bear' cannot be mean a bear that is 'Teddy', 'Teddy' is not an attribute of a bear so not [Teddy + bear]. Teddy + bear is instead a different entity with a different perceptual meaning.

3. Learning perceptual meaning: [Lar17] refers to this evaluation measure as amounting to "updating classifiers based on sensory observations of visual scenes and associated linguistic descriptions."

4. Flexibility to work with the state of the art classifiers

The ability to working with the state of the art classifiers gives a flexibility to the model to be improved. Models that classify perception simultaneously is necessary for obvious reasons.

The capacity of these models in handling linguistic phenomena such as intersective and non-intersective compostionality is very importnant in many aspects. The better the model at distinguishing the implied meaning in these two forms is good for generating more detailed descriptions for a task like image-captioning.[Nik+19]

The ability to deal with compostionality is even more crucial if we employ these models in interactive tasks, such as dialogue or question-answering. There are examples, mentioned later in this paper, where the vision-language model does not only need to classify object and give answer, but the answer requires reasoning from high-level data (text) and look if the inferential meaning is satisfied within the perceptual representation.

1.3.3 image captioning

1.3.4 VQA

Simultaneous Localization and Map Building(SLAM) is a problem where a robot should be able to map an unknown environment without a GPS or local map. Simultaneous localization is when a robot discover its surrounding and simultaneously construct a map while aware of its changing location. This means that the robot should extract information from its surrounding and learn the map as it goes.[Gri+10] [Dis+01] [Zha+18]

The robot has to discover its surrounding and simultaneously update/know where its location in order for it to construct the map.

1.3.5 EQA

Since the embodied question answering is multimodality, related works can vary between different domains. The first category of research includes the embodied-question answering task, including navigation and visual question answering. The second is research that aims to improve the VQA model, which on its own can be considered as multimodality, where the architectures of such systems aim to integrate visual and linguistic features (Images and questions).

https://arxiv.org/pdf/1712.03316.pdf

http://proceedings.mlr.press/v100/thomason20a.html

https://arxiv.org/pdf/1611.08481.pdf

1.3.6 Datasets

https://arxiv.org/pdf/1511.03416.pdf

https://arxiv.org/pdf/1505.00468.pdf

https://arxiv.org/pdf/1505.05612.pdf

https://link.springer.com/content/pdf/10.1007/s11263-016-0981-7.pdf

https://www.aclweb.org/anthology/D16-1264.pdf

https://arxiv.org/pdf/1410.0210.pdf

https://arxiv.org/pdf/1506.00278.pdf

https://arxiv.org/pdf/1505.02074.pdf

1.3.7 vision-language models

http://proceedings.mlr.press/v100/thomason20a.html

https://arxiv.org/pdf/1511.02274.pdf

https://arxiv.org/pdf/1606.01847.pdf

https://arxiv.org/pdf/1812.05252.pdf

https://arxiv.org/pdf/1603.01417.pdf

Chapter 2

Background

2.1 Habitat

The Habitat Project introduces a new task called Embodied question-answering. The EQA task consist of two sub tasks. The first is navigation, and the second is Visual Question-Answering. The agent needs to successfully navigate to the target object. Once the agent reaches the goal it would stop at the target location and would process the question and the visual input to answer the question.

The system consist of four modules. They are vision, language, navigation and question-answering. Vision and language are used for both navigation and question-answering.The vision-language modalities are fine-tuned differently for navigation and QA.

Navigation and question-answering are trained separately. Imitation learning is used for navigation, and supervised learning for question-language.

There is no available connected model that connects(Nav,VQA). The researches elaborate that the system preform poorly if the two modules put to work together. Both modules use the shortest path as a way to reach the scene of the question. However, the navigation might go off the shortest path and seek to take more actions to reach the goal. This might lead to distorted images and inaccurate view point of the object in question. "Noisy or absent views" would confuse the question-answering model. Therefore, the navigation is frozen when once it completes a navigational episode.

2.1.1 vision

The vision of the system relies on egocentric 224x224 RGB images processed in CNN. The CNN encoding has the functionality of a "multi-task pixel-to-pixel prediction framework," which consists of 4 5x5 Conv, BatchNorm, ReLU, 2x2 Max-Pool blocks, and they produce a fixed-size representation.

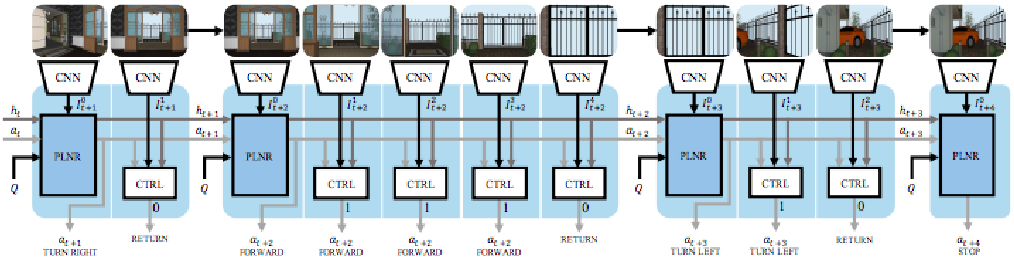
It is possible to train the encoder-decoder on generating three sensory information. The three decoders, which can also be referred to as sensors take the functionality of: 1) RGB reconstruction, 2) semantic segmentation, and 3) depth estimation. The latter sensors are used to obtain "object attributes (i.e., colors and textures), semantics (i.e., object categories), and environmental geometry (i.e., depth)."

In the baseline models, different tasks take different sensors.Not all the above-mentioned sensors are used in all the baseline tasks. Since navigation and VQA are trained and evaluated separately, we refer to them as separate "tasks". The two tasks in EQA take the following sensors:

Navigation: "depth" and "RGB". Depth sensor is essential for the agent's capability to navigate. With depth sensor it could estimate distances and avoid colliding with obstacles.

VQA: "RGB". No reason mentioned to why the other sensors are not used in the question-answering baseline module.

"The range of depth values for every pixel lies in the range r0, 1s, and the segmentation is done over 191 classes"(p.11). (page,6)



2.1.2 language model

2.1.3 navigation

Habitat's navigation is referred to as PACMAN. It consist of two core components, planner and controller. The planner takes inputs from the vision and language model, and the encoding of hidden-layer and action of the previous time-step , then outputs action-decision.

The controller takes the previous hidden state and action-decision and executes the action. A visual input is passed to the control then the controller classify the next decision of two possible decisions. Either to repeat the last action given by the planner or to return to the planner. The controller can repeat the same action maximum five times then it automatically returns to the planner.

Visualization of the navigation is in figure (1). T stands for the planner's time-steps, t = 1,2,3...., and N(t), n = 0,1,2,3.. denotes the controllers time-steps. The denotations of symbols explained clearer in the quotation :

”I_t^ndenote the encoding of the observed image at t-th planner-time and n-th controller-time. The planner is instantiated as an LSTM. Thus, it maintains a hidden state h^t (updated only at planner timesteps), and samples action $a_t \in \{forward, turn-left, turn-right, stop\}$ ”p(6)

For eample the first step-decesion from the planner is denoted as such:

$$a_t, h_t \leftarrow PLNR\,(h_{t-1}, I_t^o, Q, a_{t-1}),$$

The planner computes the next step-action a_{t+1} from input of the previous hidden layer (h_{t-1}), question encoding (Q), the previous action a_{t-1}, and the image input given to the PlNR (tI_t^o).The planner selects the action a_{t+1} and update the hidden state h_{t+1} then passes the control to the controller.

(The basis of the controller decision is a bit unclear)

The controller decides to either repeat the action or return control to the planner. The controller's classification is based on the current hidden-stateh_t and current action a_t and the image observation from the planner + the image given at the controller's time-step. The denotation of the classification is as such:

$$\{0,1\} \;\; c_n^t \;\; \leftarrow CTRL \;\; (h_t, a_t, I_t^n)$$

"if $c_n^t = 1$ then the action a_t repeats. Else $c_n^t = 0$ or a max of 5 controller-times been reached, control is returned to the planner"p(6). The h_t a_t coming from the planner act as an intent. The controller, initiated as "feed-forward multi-layer perceptron with 1 hidden layer",repeats and controls the action in order to align I_t^n with intent given by the planner.

Imitation-learning

Nav evaluation

2.1.4 VQA

2.1.5 datasets

The datasets consist of two parts. One is a 3D indoor enviroments, and the other is a question-answering deta-set. The 3D enviroments are constructed images that assimilate real indoor enviroments. The 3D Scenes and the QA dataset mentioned in [Das+18], are called SUNCG(3D houses) and "EQA V1" (QA). The EQA V1 is a synthetic dataset generated automatically, and constructed based on the setting of the 3D houses in SUNCG.

SUNCG is no longer available. [Das+18] changed the SUNCG 3D setting to MatterPort 3D (MP3D). MatterPort 3D is a reconstruction of 3D houses in (SUNCG) scene dataset. The latter also implies that the inital "EQA V1" is not applicable for MP3D.

The new QA dataset for Matterport 3D is available but not the code that generated it. The EQA.v1 is a synthetic dataset generated automatically. There is a available code to generate QA for SUNCG, but the question generator for the latest published QA for MatterPort is not available.

A few of the differences between the question dataset for SUNCG (EQA-SUNCG) and MP3D(EQA-MP3d) are mentioned in [Wij+19]. However, not in all the information in [Wij+19] seems to match with EQA-MP3D that we have. In [Wij+19] page(4) its stated that the number of scene used from MP3D is 76. The dataset we downloaded from "facebookai/habitat" repo on github uses a total 67 scene of 90 scenes available in MatterPort3D. 57 of the 67 scenes are used for questions in the train-set and 10 in the the enviroment. Note that the latter implies that the robot is tested on different scenes from the scenes it has been trained in.

We refer to each question-sample in the EQA data-set as an "Episode". Each question is an episode, because the sample contains also, on the topic of the question-info, geometric information and shortest paths. Each episode is applicable for navigation and VQA, and can be run for each task separately.

The EQA-v1 dataset consists of 1950 validation sets and 11000 training questions.

Scene Dataset

(restructuring is required– more precision) (examples to rephrase– why do we need the location in global coordinates and why the camera views are also important)

The MP3D dataset provide 90 segmented houses with their semantic annotation. The semantic annotation is segmented based on the structure of the house. The segments consists of house levels(floors), to regions(rooms), and objects. The annotation is organized accordingly, such as that an object is annotated and indexed in relation to room and the floor its located in.

For example, the annotation of a house begins with the first level in it, followed by the rooms and objects in each room as: house 1 [level1:room1[bedroom]:(obj1:bed,obj2
level2:......]

Each semantic annotation include geometric information. The geometric information consist of elements as location of an object, region or level, defined by their center in a world coordinate system. Other information is the size of the entity given its radius from its starting location (center).

The camera views of the scenes are globally oriented [Cha+17](p3). A way to allocate an object is to find its location in a accordance to global coordinates. Let's say the global coordinates start from the center of a house where the center of the house is (0,0,0) on the (x,y,z); and let's say all the objects are spawned through out the house's (x,y,z) axis where each objects location is defined by its distance to the house center. When annotated, the objects are viewed through a camera. The description of their geometric location, thus, should consider the view-postion of the camera.

In graph (A) in figure 3, we see that the camera-view of coordinates align with the global coordinates.The (x,yz) that go through each object in graph(A) and graph (B) are the view of the axis in reference to the camera. However, if the camera is positioned to the right of the object from our view, as in graph (B), then we say that the camera view of coordinates is not aligned with the global view. We notice in graph (B) that from the camera view, the "global X" is "Y" and vice versa.

Some geometric calculations cannot be preformed if the location measurements are not relative to each other. For example, if we want to calculate the distance between objects the locations must be consistent with one reference point. The camera position is changing and if the location of an object is referenced by the camera's position then we would get locations relative to the changing position of the camera in a time-span.

To globalize the orientation of the view, measures such as top-down view of a map, or calculating the rotation of the camera from the global center. While the global locations are crucial for measuring the distance, other point-views are also crucial for other purposes. There are three essential coordinate systems to know when working in a 3D environments:

1. World coordinates(global): World coordinates(global): The coordinate system that starts at the center of the world; a house in our example. The center of an object in this coordinate system, is then decided by its distance to the center of the world.

camera-view coordinates: The coordinates from the camera's views. The center of this coordinate system is the position of the camera. The center of the object in this world is defined by its distance to the camera.

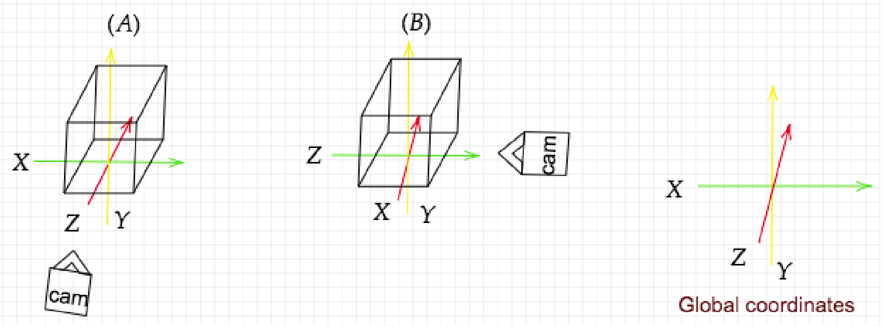
3. Local view: The center of the local view is the object itself.

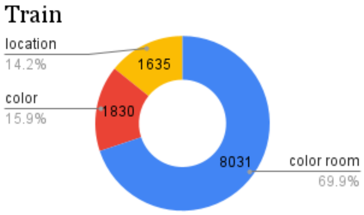
The center of all these views is (0,0,0). We described above that the world coordinate system allows us to measure distance between objects in a world map. The camera view is useful if a robot is expected to navigate an environment and describe spatial relations between objects such as "next to", "above". The local view could tell about the size of an object. In particular, the (x,y,z) from a local point of view tell about how far the object stretches from its center where the center is (0,0,0). The local view can be referred to as "radius".

MatterPort 3D provide the views decribed above. We discuss in more detailed the usage of the object's location in global coordinates and the local view in details in the implementation part.

Task Dataset

(More information to include– 1. How they filter out questions based on entropy, and how they filter out objects based on size..2.How many unique question there is. 3. Explain more thoroughly how the singleton(object,room)



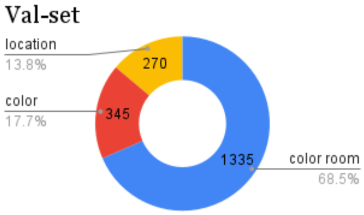


works)

The question-answer data-set contains three types of questions. Each question in the detest is a function that can be executed in the environment to give an answer. More in section (3.2)

(To include the number of unique questions here) There is a total 11496 questions in the train split and 1950 questions in the val split. As seen in figure (5),in the train split there are 1830 questions "color" type, 8031 of "color room" and "1635" of location type. For the validation split there are 1335 "color room" questions, 345 "color" questions, and 270 "location" questions.

Each question-type is generated in a string template.The templates



are as the following:

- color room template: ”what color is ¡obj¿ in ¡room¿?”: In these questions the agent needs to find the room in question and look for the object and answer the question. For the agent’s to be successful at reaching its target, it needs to know the difference between rooms, and objects, as by implicitly recognizing that a certain room is a living-room, not a bathroom and such.

- color template: ”what color is ¡obj¿”. The difference between ”color” type and ”color room” is that no room is specified in the ”color” type of question. In ”color” type the agent needs to figure out where to look by itself. For example, ”what color is the fridge?”, the robot needs to implicitly figure that the fridges are usually in the kitchen and navigate to the kitchen to answer the question. In other cases, the object could be in the vicinity of the robot’s starting point, so that it all it needs to do is to look around.

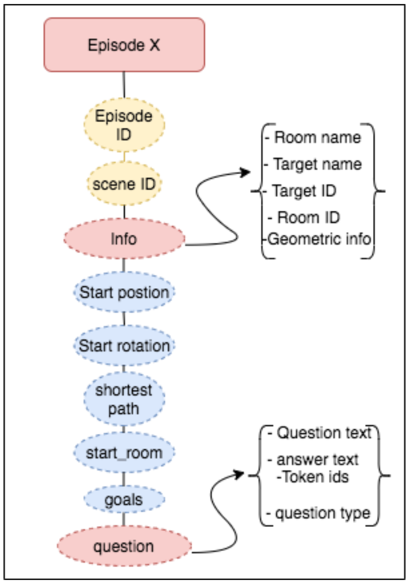
- location template: ”What ¡room¿ is the ¡obj¿ located in”.

Querable objects and rooms The questions ask about 50 unique objects. [Das+18] in (page 4) describe the process of object and room selection for question generation.The following quoted from (page 4):

”select(objects)-¿singleton(objects)-¿query(location)”

The above represents the steps taken for finding object and location to fill in the questions template. select(objects) is a function that collects all the objects in the house. singleton(objects), filter out an object that occurs only once in the house; query(location) finds the location of the object.However, this applies to the old dataset in SUNCG.

In EQA-MP3D, the object in question is not unique to the house but to the room. The latter means that for an object to be selected for a question, there need to be only one instance of that object existent in the room. The reason for this is to avoid ambiguity, and not to confuse the agent if there happen to be more instances of the same object in the room.



we observe that all the objects that the robot is asked about in testing have occurred in the training questions. While it has been mentioned earlier that the robot is tested in different scenes from the scenes it was trained on, similar objects from the training co-occur in the testing. The latter means, in particular, that the robot is unfamiliar to the test scenes but familiar with all the objects that are being asked about in the test. This information is also stated in [Wij+19].

Structure In figure (x) we see the top structure of the val and test. *Episodes* refer to each question-function in the data-set split. *Question vocab* and *answer vocab* contain the same elements as dictionary keys. The elements are: [word list,stoi,itos,num vocab,pad token].

”Question vocab” and ”answer vocab” in the ”train” and ”val” are identical to each other. When using each split of the dataset, the answer-tokens that are considered are the ones contained within the episodes instead of the word-lists mentioned above.

Each question-sample is an episode that consist of multiple layer information. The structure of one episode of all the ”episodes” is as seen in figure(x). We describe the elements of an episode in the following:

House ID: The house ID given by the house ids in MatterPort3D. **Episode ID**: The episode index in the range of the split's length. **Info**: This element contains all the information about the the object and room in a question. The information is structured as such:

Information about the traget-object is the first layer within "info":

centroid: The center of the object's box in the global coordinates. Box is the area that labels the object. When the center is globally oriented we would refer to this center and box as Axis-aligned bounding box(AABB), which means that (x,y,z) axis of the center are aligned with global coordinates.

radi: It tells how far the box (object) stretches from its center one direction of each axis. The value of radii is relative to the object itself (from the local view), where the center is zero. If we have, for example a radi of (2,1,4), this means that the object's box stretches +2 and -2 from the center on the x axis. The boundaries of the object's box relative to itself is referred to as object oriented bounding box (OOB).

level: at which level-floor of the house is the object located in.

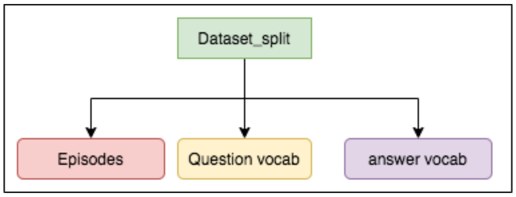
room-id, *room name*,*obj Id*,*room name* : Room ID, room name and object ID as given by semantic annotation in Matterport3D.Many of the objects are re-named, mostly names in hyponymes changed to hypernym category such as: round-sofa, l-shaped sofa changed to their hypernym category "sofa".

The second layer is information about the room:

Information about the room is similar to the type of information given for the objects. Th information is *floor-level*, *room-id*, *room name*,

Final layer consist of a "question-meta" which includes the color of the object. This section also includes question-entropy .....

The elements that are marked in blue in figure(x) are navigation-related material.



start position: The start positions are all unique. For each unique question in the data set there is fifteen different starting position.

rotations: This is the rotations that the agent have to do while navigating. It stands as supplementary information for the shortest path

goals: Goals are the destinations that the agent should reach in navigation. The goals stand for the possible view points from where the the target object can be looked at by the robot. Each view point consist of geometric position and the rotation toward the target object respective to the position.

2.1.6 Bias and answer distribution

Evaluation

Baselines

expirement The idea is to extend the question asked for the agent. The two types of questions are size and spatial. The process of question extension includes using information from the initial EQA-v1 dataset, which consists of color, color-room, and location questions. Each question sample has a target object with corresponded information as object ID, room ID, Scene ID, question(token-ids and text), and shortest path. We pick the object and the room ID for every question sample to extract the rest of the information about the other things in the room. The extracted information is the volumes of the objects and the

Chapter 3

implementation

3.1 Extending Dataset

3.1.1 house parsing

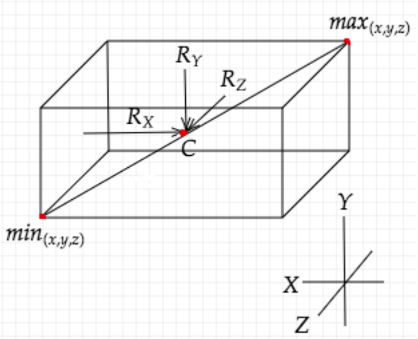
(Parsing house annotation will be changed from the house files in matterPort to the parser of Habitat simulater. Habitat simulator provide more information in a better categorization such as sizes of aabb and obb. In House files mp3d only the size of oobb given. Also object names in habitat simulator are alignining with the object names give in the eqa dataset)

Each house environment comes with three files. The three files are x.house,x.ply and x.. We collect the annotations from the x.house files house.

Each house file comes with eleven line-types of annotations.. The lines are marked by a capital letter as a marker; the first letter-marking to the last letter are as in this list [H,L,R,P,S,V,P,I,C,O,V]. Each letter-marker symbolizes a certain type of information. In this section, I am going to explain only the type of information that we use in this project.

The only data we extract from the house file, is the "O". The "O"

[0] https://github.com/niessner/Matterport/blob/master/data_organization.md



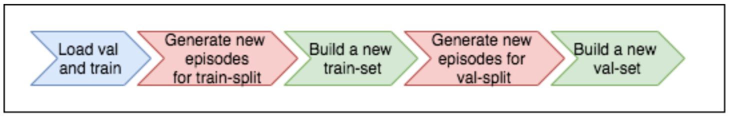
lines contain information about the objects in the house. Every line that begin with an O letter consist of one object in the house with a corresponding information about its geometry and location within a room and level-floor. Each "O" line looks as such: [O object_index region_index category_index px py pz a0x a0y a0z a1x a1y a1z r0 r1 r2 0 0 0 0 0 0 0 0]

The data of the object in the line seen above comes in a string form, and each section in the string represents different types of information. *Object_index*, the index of an object is what we refer to as the object ID. *region index* is the room ID. *category_index* is the object's index in category map; this index is used to obtain the object's name from the category map.*px py pz* represent the center of the box in (x,y,z) axis. *a0x a0y a0z a1x a1y a1z* these are the rotation of the OOBB and AABB. *r0 r1 r2* represent the radius of the object from the center on the (x,y,z). Finally the last "0"s in the line have no meaningful value, and therefore are ignored.

Extracting data and structuring

We extract two types of raw information from each object's line of annotation. First we take the obj and room indexes (ids). Second is the [px py pz] and where we categorize it as the center of the object's box. Third is the [r0 r1 r2] (radius-half-extent).

We make calculations from the data we extract in order to obtain other necessary info for generating question. The first calculation is finding the 'min' and 'max' of a box given an object's center and half-extents(r). In


Load val and train
Generate new episodes for train-split
Build a new train-set
Generate new episodes for val-split
Build a new val-set

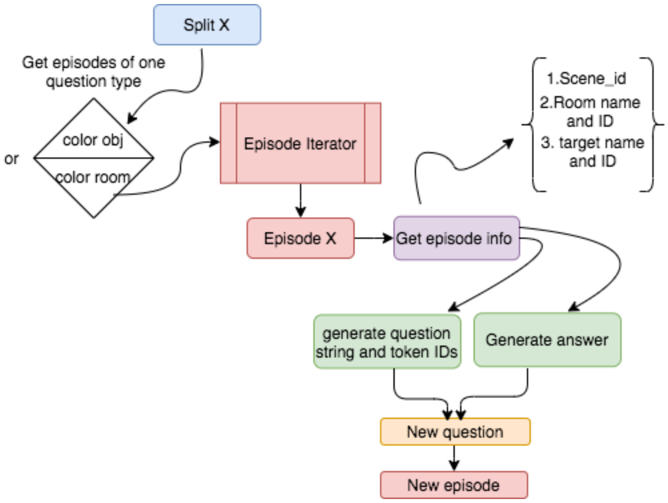


figure (X) we see visual representations of the extracted and calculated data. The min represents the corner of a box that has the lowest value of (x,y,z) and max is the corner with largest value for (x,y,z). Other way to put it, the min represents the corner in the minus direction from the center in all the axis, and max is the corner on the positive direction from the center in all axis.

We subtract and add the half-extents with the center value. Min = (c)

Our house parser consist of two classes. The first class is a class that parses the houses into a structural data. The second is functional class we use to find near objects close to a target object. The latter class is used

3.2 question generation

3.2.1 Filtering objects

3.2.2 size-questions

3.2.3 spatial-questions

Chapter 4

Evaluation