Learning Embeddings that Capture Spatial Semantics for Indoor Navigation

Vidhi Jain
Carnegie Mellon University
vidhij@andrew.cmu.edu

Shishir Patil
University of California, Berkeley
shishirpatil@berkeley.edu

Prakhar Agarwal
University of Washington, Seatle
pa2511@uwashington.edu

Katia Sycara
Carnegie Mellon University
katia@cs.cmu.edu

Abstract

Incorporating domain-specific priors in search and navigation tasks has shown promising results in improving generalization and sample complexity over end-to-end trained policies. In this work, we study how object embeddings that capture spatial semantic priors can guide search and navigation task in a structured environment. We know that humans can search for an object like a book, or a plate in an unseen house, based on spatial semantics of bigger objects detected. For example, a book is likely to be on a bookshelf or a table, whereas a plate is likely to be in a cupboard or dishwasher. We propose a method to incorporate such spatial semantic awareness in robots by leveraging pre-trained language models and multi-relational knowledge bases as object embeddings. We demonstrate the performance of using these object embeddings to search a query object in an unseen indoor environment. We measure the performance of these embeddings in an indoor simulator (AI2Thor). We further evaluate different pre-trained embedding on *Success Rate* (SR) and *Success weighted by Path Length* (SPL).

1 Introduction

Consider an example of finding a key-chain in a living room. A key-chain can be generally found either on a coffee table, inside a drawer, or on a side-table. When tasked with finding the key-chain, a human would first scan the area coarsely and then navigate to likely locations where a key-chain could be found, for example, a coffee table. On getting closer, they would then examine the area (top of the table) closely. Following this, if the key-chain is not found, they would try to navigate to the next closest place where the key-chain could be found. In all these scenarios, the presence (or absence) of a co-located objects would boost (or dampen) their confidence in finding the object along the chosen trajectory. This would, in turn, influence the next step (action) that they would take.

Our goal in this work is to enable embodied AI agents to navigate based on such object-based spatial semantic awareness. To do this, we focus on the following problems: a) training object embeddings that semantically represent spatial proximity, and b) evaluating these embeddings on semantic search and navigation tasks. The embeddings we learn should capture the following - a) learn about larger objects around which the smaller items could be found (e.g., key-chains are likely to be found on / near tables), and b) learn about objects that are found mutually close to each other, e.g., (key-chains and credit-card). By learning such embeddings, we want to capture the semantic relations in terms of distance. We train embeddings that capture the spatial semantics using a multi-relational knowledge graph. Further, we formulate an algorithm to compare the performance in terms of *Success Rate* (SR) and *Success weighted by Path Length* (SPL) across different kinds of embeddings. Interestingly,

NeurIPS 2020 Workshop on Object Representations for Learning and Reasoning, Vancouver, Canada.

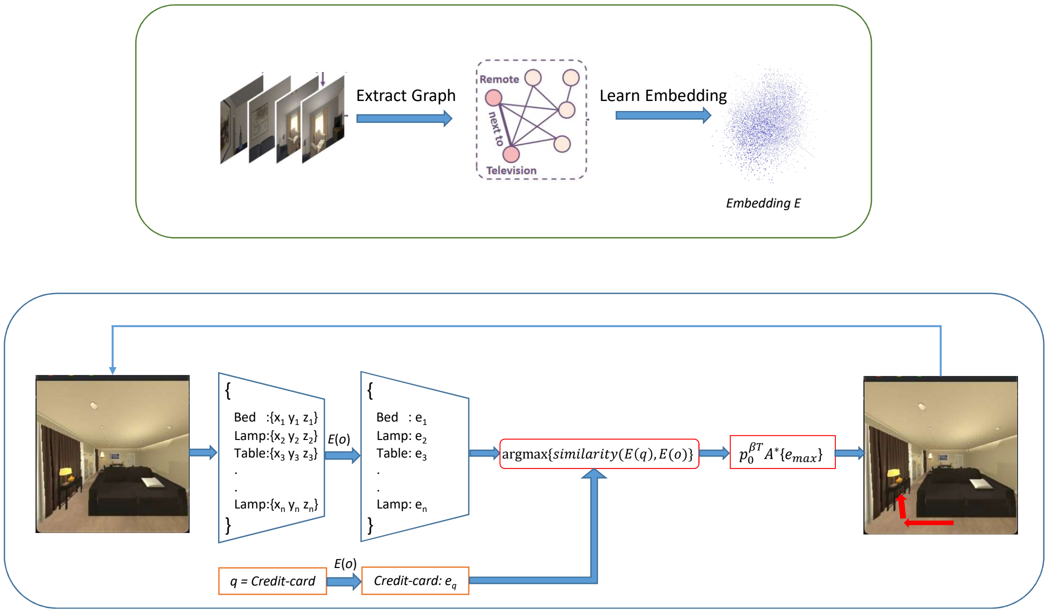

Figure 1: [Top] Training the embeddings, [Bottom] Using embeddings for navigation

embeddings trained with general-purpose text corpora like Word2Vec [1], FastText [2] provide a strong baseline for our task. We demonstrate our techniques on the AI2Thor [3] environment.

Our contributions are summarized as follows: (1) We learn and contrast representations to capture spatial semantics. (2) We demonstrate that a momentum inspired, similarity based greedy navigation technique results in success rate $> 90\%$. (3) Finally we show that using spatial semantic prior knowledge can significantly improve the navigation performance.

2 Related Work

Classical navigation techniques in robotics generalize to unseen environments by mapping, localization and path planning. However, most of these approaches fail to leverage the semantic structure of the environment, like certain objects are mutually close to each other.

Multi-relational knowledge-base based embeddings such as RoboCSE [4] have demonstrated the ability for an agent to predict object affordances, and materials in the real world. The RoboCSE were trained using the object metadata from AI2-Thor simulator. While these embeddings were used for prediction of object affordances by an immovable agent, we differ in our approach of utilizing these embeddings by evaluating how well they capture the spatial semantics for query search and navigation task for a mobile agent.

Prior work has also looked at using object embeddings for robot navigation. Graph convolutional networks have been used for learning neural network based policy in [5], and [6] for object search and navigation. As these are data-driven approaches, they require several interactions in the environment to learn the policy. We propose a sample-efficient approach based on pre-trained embedding [Sec 4]. Unlike learning a navigation policy from scratch, our approach shows promise of transfer to the real-world settings.

Recent work in object navigation [7] shows a modular approach to map the environment and predict sub-goal based on the semantic object map. We do not address the vision based mapping as addressed in this paper. Instead our approach is a modular component that can be integrated with classic robotics paradigm of mapping and planning, as we focus on high-level navigation decisions aligned to spatial semantics of objects.

Algorithm 1: Navigation towards query for the objects in the field of view

```
Result: Plan to the query object
embedding module E;
visited = [];
while query q not found do
    for all objects o in (360 degree) field of view do
        if object o not in visited then
            scores = similarity($E(q), E(o)$);
            Add $o$ to potential_sub_goals;
        end
    end
    sub-goal = potential_sub_goals[$\arg\max$ scores];
    proposed plan $p_0^T := A^*$ search to sub-goal;
    move according to initial part of the proposed plan $p_0^{\beta T}$ where $\beta < 1$;
end
```

3 Task Definition

Our goal in this work is to navigate an agent from a randomly initialized location in a scene to a specified query object. The task is considered a success if the agent can view the object of interest and is within a small distance (~ 1 units $= 5 \times 0.25$) of the query object.

Consider the agent is tasked to find a *credit-card* (query). The agent is initialized randomly in a living room and records all the objects in its 360 degree field of view. By design, we enforce that the agent can only see “large” objects. Imagine you enter a living room and see sofa, table, TV, etc easily. However, depending on your distance, the objects such as the credit card might be occluded or not be visible. We consider large objects that occupy more than 1% of the total screen size of the camera’s image. This threshold is analogous to the effects of a real camera as detection of objects in the frame would depend on the camera resolution.

The agent now finds a sub-goal. The sub-goal is the most likely (large) object in its field of view, which we expect is close to the query object, or could lead the agent towards a viewpoint that brings relevant sub-goals in the agent’s field of view. This is done by finding the pairwise similarity of the query object’s embedding with the embeddings of all the sufficiently large objects in sight. For example, out of visible objects as table, lamp, and bed, we would expect embeddings of a table to have highest similarity with the embeddings of a credit-card.

Once a sub-goal is chosen, we calculate the potential plan $p_0^{T_i}$ where T_i is the total steps required to reach the sub-goal i. The agent executes $p_0^{\beta T}$ that is some fraction of the initial part of the potential plan. The agent then looks for any new object that is now visible, and re-ranks the similarity of all objects visible to it to find the new sub-goal. This process repeats until the query object is found. Consider that in a 1D space, the agent’s location is at 0, and the query objects location is at 10. With $\beta = 0.5$, the agent would first navigate to point 5, then to 7, and finally to 9 before reaching 10. This approach has two advantages: a) improves the performance of the system by exponentially reducing the amount of compute per navigation (scan, similarity computation, A^* path computation). b) helps overcome oscillation. Oscillations in navigation happen when the agent takes a step towards sub-goal A, and finds B to be having a higher similarity. On shifting the sub-goal, and taking a step towards B, B becomes occluded, and the sub-goal rotates back to A.

4 Methodology

Our novel approach to find the semantic similarity between the given query and visible objects is formulated based on the distribution of distance between a pair of objects. Additionally, we identified two broad kinds of embeddings for our analysis, as discussed below. Each of these embeddings are evaluated for navigation as outlined in the algorithm 1.

			SPL by shortest path length l		
Room Type	Embedding Type	SR/SPL	l< 10	10 <l< 20	l >= 20
Kitchen	Graph Embedding	**0.992/0.639**	0.644	0.642	0.644
	RoboCSE	0.960/0.624	**0.650**	0.643	**0.867**
	FastText	0.983/0.615	0.626	0.624	0.573
	Word2Vec	0.984/0.626	0.633	**0.649**	0.581
Living Room	Graph Embedding	0.919/0.692	0.777	0.698	**0.693**
	RoboCSE	**0.942/0.686**	0.766	0.692	0.614
	FastText	0.908/0.682	0.774	**0.727**	0.619
	Word2Vec	0.908/0.666	**0.793**	0.708	0.596
Bed Room	Graph Embedding	0.954/0.678	**0.739**	0.631	0.659
	RoboCSE	**0.966/0.681**	0.731	0.628	**0.690**
	FastText	0.960/0.686	0.738	**0.657**	0.544
	Word2Vec	0.956/0.662	0.720	0.624	0.576
Bath Room	Graph Embedding	0.997/0.694	0.692	0.733	-
	RoboCSE	0.994/0.692	0.693	0.716	
	FastText	0.994/0.688	0.692	0.690	
	Word2Vec	**0.998/0.701**	**0.697**	**0.743**	

Table 1: **Results using termination (stop) action.** Success Rate (SR) and Success weighted normalized inverse path length (SPL) for different floor plans in AI2thor environment; l denotes the shortest path length to query object.

Pre-trained word embeddings Language embeddings have been shown to capture the word-level semantics in their metric space. For example, FastText embeddings have shown promising results for semantic navigation in procedurally generated environments [5]. To understand if these embeddings can be transferred in terms of object semantics for the downstream task of navigation , we test our algorithm on the Word2Vec [1] and FastText [2] embeddings.

Knowledge base embeddings Multi-relational embeddings encode abstract knowledge that could be obtained by the agent from its sensors or an external knowledge graph. RoboCSE [4] uses ANALOGY[8], a semantic matching method, to learn multi-relational embeddings processed from the AI2Thor environment data, where the nodes represent the objects and edges denote the relation between them. Further, we also learn a Graph Embeddings by treating all relations as being equivalent and using DeepWalk [9] to learn embedding for the resulting undirected graph.

5 Experiment and Results

We use the AI2Thor environment to extract pairwise object relations to train the embedding network. We make two interesting design choices that we think will be valuable to the community. First, since we assume perfect object detection if the object is sufficiently in the field of view, we determine sufficiency as a parameter with respect to the percentage of pixels occupied by the object in the field of view. This parameter tries to replicate the variation in detection arising from camera quality used on an agent and noise in object recognition. Second, we adaptively choose step-sizes to navigate towards the sub-goal. This allows the agent efficiently to look for other objects in the field of view on the way.

In Table 1, we show the performance of different embeddings for navigation task in terms of Success weighted by normalized inverse path length (SPL) metric. Further, we also report SPL computed by grouping cases where the shortest path length to the query object is greater than a particular threshold. For example, the credit card may be initialized (randomly) 15m away from the agent; so we would include it in $10 < l < 20$ bucket.

We observe that the graph based embeddings (RoboCSE, and Graph Embeddings) perform similar to language based embeddings (Word2Vec, FastText) in those cases where the optimal path to the target object is $l < 10$, but outperforms the baselines when $l >= 20$. This is because, if the query object is close by ($l < 10$), any action that the agent takes has a higher probability of heading towards the query object. However, for objects that are further away, the agent has to find the optimal path to get to the query object.

Broader Impact

Our approach has been to learn an embedding that encapsulates the relationship between the smaller query object (like an apple) and their possible parent receptacles (like refrigerator). While this work in-itself does not significantly change much, we believe a more extensive system built with the principles defined here would enable efficient and more straightforward techniques for finding navigation goals. The failure of the system might lead to oscillations (agent navigating halfway to object A, then halfway to B, and back to A...). Lastly, we believe that our treatment of pre-trained embeddings is by no means comprehensive or exhaustive.

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0036. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[3] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. 2020.

[4] Angel Daruna, Weiyu Liu, Zsolt Kira, and Sonia Chernova. Robocse: Robot common sense embedding, 2019.

[5] Niko Sünderhauf. Where are the keys? – learning object-centric navigation policies on semantic maps with graph convolutional networks, 2019.

[6] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *CoRR*, abs/1810.06543, 2018.

[7] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration, 2020.

[8] Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. *arXiv preprint arXiv:1705.02426*, 2017.

[9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.