

Playing with Embeddings : Evaluating embeddings for Robot Language Learning through MUD Games

Anmol Gulati* and Kumar Krishna Agrawal*

anmol01gulati@gmail.com, kumarkagrawal@gmail.com

Indian Institute of Technology Kharagpur

Abstract

Acquiring language provides a ubiquitous mode of communication, across humans and robots. To this effect, distributional representations of words based on co-occurrence statistics, have provided significant advancements ranging across machine translation to comprehension. In this paper, we study the suitability of using general purpose word-embeddings for language learning in robots. We propose using text-based games as a proxy to evaluating word embedding on real robots. Based in a risk-reward setting, we review the effectiveness of the embeddings in navigating tasks in fantasy games, as an approximation to their performance on more complex scenarios, like language assisted robot navigation.

1 Introduction

Language provides a natural interface for humans to communicate with robots. With their increasing public presence, from self-driving cars and rescue operations to warehouses, it is imperative to reduce this barrier of communication, by improving language learning in mobile robots. For instance, in search and rescue operations one might want to instruct the agents to "Reach the third floor of the blue building". Given the nature of the tasks, it is essential for the agent to accurately infer its current state and parse natural language instructions to corresponding actions.

In recent years, learning continuous representations of words and symbols have become central to dealing with several key problems in natural

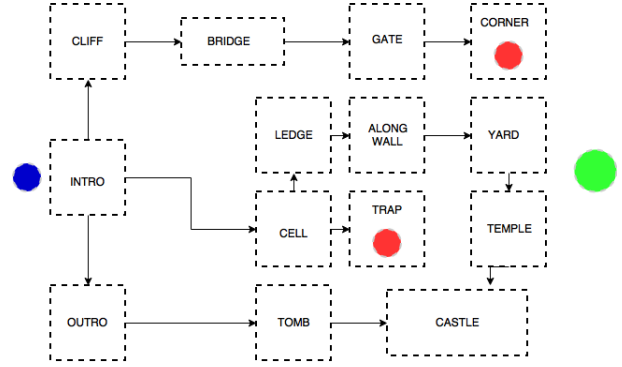


Figure 1: An example of environment generated in the MUD environment. While the blue blob denotes the start state, the red blobs denote deterministic intermediate negative rewards. The green blob on the right denotes positive reward on completion of quest.

language understanding. Usually trained to maximize the likelihood of next utterance, these models effectively learn co-occurrence statistics based on corpora, motivated by the distributional hypothesis (Harris, 1954). This objective function, often organizes objects and actions with *similar* semantic information, to close neighborhoods in the corresponding embedding space. Here, the *similarity between words* is often defined by some metric for *similarity* of the word vectors.

Though evidently successful across several tasks, common general purpose word embeddings with fixed dimensions are often inadequate in their representation capacity. In recent works, (Gauthier and Mordatch, 2016; Lucy and Gauthier, 2017) argue that language learning grounded in perception, motion or actions, are important to learn meaningful representations corresponding to *consequences* of actions on objects. Specifically, in a scenario where robots are deployed in real world, it is important to distinguish even between

*Equally contributed to the project. Author names listed in alphabetical order.

Fantasy Environment

You are standing very close to the bridges eastern foundation. If you go east you will be back on solid ground ...
The bridge sways in the wind.

MARCO dataset (Chen and Mooney, 2011)

face the octagon carpet. move until you see red brick floor to your right.
turn and walk down the red brick until you get to an alley with grey floor. you should be two alleys away from a lamp and then an easel beyond that.

Table 1: Examples of instructions from MUD vs MARCO dataset for navigation.

similar actions based context and their effect in a given situation.

In this paper, we address this task of evaluating word embeddings for language learning in robots by introducing text-based MUD games (Curtis, 1992) as a proxy. Traditionally in robotics, elaborate simulators have been used to evaluate motor control policies of motion planning for the physical agents. In a similar spirit, MUD games provide a rich playground for define complex scenarios solely with textual descriptions. Based on these, the agents are required to take certain actions with the objective of clearing the quest. (Fig. 1)

To evaluate, we train a LSTM-DQN following (Narasimhan et al., 2015), in a reinforcement learning framework. The agent is trained to learn control policies, with the objective of maximizing the rewards obtained. We demonstrate that using general purpose embeddings improve on the average rewards per quest.

2 Multi-User Dungeon Games

A Multi-User Dungeon Game, as defined in (Curtis, 1992) *”is a network-accessible, multi-participant, user-extensible virtual reality whose user interface is entirely textual”*. In essence, it describes an elaborate environment with multiple characters and tasks for the player, the only constraint being that all interactions with the game are purely textual.

In (Amir and Doyle, 2002), the authors elaborately elucidate the richness of such environments, with insights specific to robotics. Primarily in our context of robot language learning, we make the following observations :

- The environment is partially observable, with

stochasticity depending on the actions taken by the agent. This inherently allows us to cast the problem as an MDP, and studying it in the light of recent advances in Deep Reinforcement Learning.

- Intermediate rewards provide feedback to the agent, as proxy for a human in a real-life setting. This allows the agent to learn from consequences and formulating the problem in a risk-reward setting.
- Language based symbolic interface for interaction with the environment. Since language remains consistent across simulator and real world, we assume that the text-world is an accurate replica of the real environment.

Among others, the above characteristics suggest that MUD games provide an ideal sandbox for evaluating word-embeddings for robot language learning.

3 Dataset

We propose using text-based MUD games as proxy for evaluating language learning in robotics. As illustrated in (Table 1) the nature of instructions generated by the game environment are quite similar when compared to real datasets, in this case (Chen and Mooney, 2011).

In this work, we use the Evennia¹ game environment, an open-source library to build online textual MUD games. Evennia allows users to create complex environments with elaborate textual descriptions, by simply writing a batch file to describe the objects, actions and possible interactions in the environment. Compared to other available datasets for navigation, the MUD environment provides a risk-reward scenario, where the environment can also be designed to have deterministic negative feedback to represent undesirable outcomes.

To provide feedback on the action taken, the agents are provided positive/negative rewards depending on the state of the game. In case of quest completion, the agents are provided a large positive reward, while predefined bad intermediate checkpoints like colliding with walls, or falling off bridges are penalized with negative rewards. Building on extensive literature in reward shaping in the Deep Reinforcement Learning framework,

¹<http://www.evennia.com/>

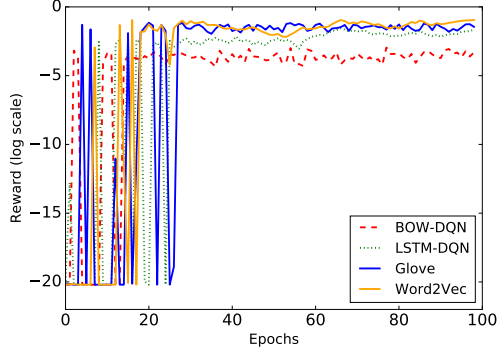


Figure 2: Evaluating agents trained across various embeddings based on average reward (log scale). Pretrained word-vectors outperform random initialized LSTM-DQN and BOW-DQN.

more complex environment can also be easily constructed to evaluate the generalizing capability of the embeddings.

4 Model Architecture

Formulating the problem in a reinforcement learning setting, we train the agent to learn a good Q-value function $Q(s, a)$, for all possible state-action pairs. More concretely the agent iteratively optimizes the Bellman objective

$$Q_{i+1}(s, a) = E[r + \gamma \max_{a'} Q_i(s', a') | s, a] \quad (1)$$

where, γ is the discount factor. With the changing distribution on Q-values the agent decides action a to maximize the future rewards.

In this regard, we follow the LSTM-DQN model by (Narasimhan et al., 2015) which builds on recent developments in Deep Q-Learning (Mnih et al., 2015).

4.1 LSTM-DQN

The input to the agent is the textual description of the current state of the environment. However, the model is required to keep a track over previous states, as well generate a compact representation for the same. Therefore we define a representation generator (ϕ_R), which is an LSTM module to keep a track of the current state of the environment. The individual time step rollouts are further aggregated (ϕ_A) to convert the textual description into compact vector representation (s).

Following the context, the model is required to choose an *action*(a) to be taken, and an *object*(o)

on which the action is applied in the MUD environment. As a result, we define two Q-value functions $Q(s, a)$ and $Q(s, o)$, which share the same state, and generate distributions on actions and objects respectively. This (a, o) tuple, together defines the action executed by the game environment to update it's state. For detailed description of the model please refer (Narasimhan et al., 2015).

5 Evaluation

We experiment on the *Fantasy World* following the environment in (Narasimhan et al., 2015). The vocabulary consists of 1340 unique tokens, with 100 different descriptions for the room. Visiting the room provides random sequence of description as developed by the designers. Average length of descriptions in the rooms was 65 words per descriptions. The possible actions per state average to 222 per state. Please refer (Narasimhan et al., 2015) for elaborate game statistics.

To evaluate the performance of agents when using different word-embeddings, different metrics could be defined depending on the requirement of the task. In the particular case of evaluating the performance of the agent on navigation task, we follow common defined metrics as in (Narasimhan et al., 2015)

- cumulative reward per episode averaged over the number of episodes.
- fraction of quests successfully completed by the agent.

Considering the task as an downstream proxy for the actual environment, this does not evaluate the individual embeddings by similarity metrics, rather the generalization across environments. In this work we compare word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) general use embeddings against a BOW-DQN baseline and LSTM-DQN trained with random initialization (Fig 2). The baseline here (BOW-DQN) uses a simple bag-of-words model to represent the textual description.

We observe that using pre-trained embeddings generally improve the performance of the model. The rewards in our primary models have high variance due to the stochastic nature of the policy. In further work, we wish to build on recent advancements in Deep Reinforcement Learning literature, for training better models.

6 Discussion and Future Work

Using continuous vector representations of words for language learning in mobile robots poses the question of choice of embeddings which capture task specific semantics. In this work we propose using MUD games as proxy for this task, rather than elaborate testing on robotic platforms.

Similar to real world situation, all interactions between the agent and the environment are through language, without any other supervision. The rewards provide the models with required supervision of tuning the word embeddings to complete the quest. Preliminary evaluation on standard embeddings show that average rewards for pretrained embeddings are better than BOW models and randomly initialized LSTM embeddings.

In future work we wish to extensively evaluate the model on more standard word-embeddings and against real datasets to establish empirical correlation between the proposed proxy and actual task.

References

- Eyal Amir and Patrick Doyle. 2002. Adventure games: A challenge for cognitive robotics. pages 148–155.
- David L. Chen and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*. San Francisco, CA, USA.
- Pavel Curtis. 1992. Mudding: Social phenomena in text-based virtual realities. *High noon on the electronic frontier: Conceptual issues in cyberspace* pages 347–374.
- Jon Gauthier and Igor Mordatch. 2016. A paradigm for situated and goal-driven language learning. *arXiv preprint arXiv:1610.03585*.
- Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162.
- L. Lucy and J. Gauthier. 2017. Are distributional representations ready for the real world? Evaluating word vectors for grounded perceptual meaning. *ArXiv e-prints*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.
- Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation.