Informing Action Primitives Through Free-Form Text

Nancy Fulda, Ben Murdoch, Daniel Ricks, David Wingate
Brigham Young University
nfulda@byu.edu, murdoch@byu.edu, danielricks@byu.edu, wingated@cs.byu.edu

Abstract

Autonomous agents require a grounded understanding of language, and multi-modal embeddings seem like a logical way to accomplish this. But once you have a joint embedding space, what is it good for? How do you use it? And most especially, how do you map human input (such as verbal instructions) into a set of behaviors that conform with the human's desires? This paper presents a method for using distributional sentence representations to prioritize action primitives. Inspired by the human ability to transfer domain knowledge via spoken language, we provide our agent with natural language instructions that describe high-reward behaviors. The agent then uses these instructions to align its actions with human intents. A key advantage of this approach is its flexibility: once a properly-structured embedding space has been obtained, new mappings between instructions and behaviors do not have to be trained using thousands or tens of thousands of labeled examples. Less than twenty data points is sufficient.

1 Introduction

This paper presents a method for imbuing agents with high-level domain knowledge via short command phrases given by a human designer. This human guidance is provided at design time and describes the behaviors a human deems necessary in order to succeed at the current task. However, in order to fulfill these goals the agent must be able to link natural language instructions with action primitives that facilitate them. Previous researchers have proposed using multimodal embedding spaces to map from images to text and back again [13, 2, 1]; however, we argue that simple conversion from one modality to another is usually insufficient to accomplish complex tasks. A multimodal embedding space that maps the word 'ball' and an image of a ball to similar locations may enable an agent to convert from natural language text to visual imagery and back again, but how shall the agent choose behaviors that comply with specific instructions such as 'place the ball on the table'?

We address this question by representing the agent's sensory input, human guidance, and action primitives in a shared embedding space - in this case, as 4800-dimensional skip-thought vectors. Our current implementation relies entirely on text-based inputs and outputs, but this method is also applicable to embedding spaces that use a joint representation to encode a wide variety of sensor inputs, motor activations, kinematic positions, and object categories. The foundational premise is as follows: sensory input, human guidance, and action primitives are all represented as points in a shared embedding space. Because this embedding space was pre-trained using large numbers of examples, it contains valuable semantic structure. We utilize this structure to leverage a small (<20) set of example pairings that map human guidance to action primitives. This mapping, which is essentially a vector, is then used to rank candidate actions based on how well they conform to the provided instructions.

We apply our method in a setting intended to mimic real-world constraints. Rather than allowing the agent to try as many actions as necessary until it accomplishes its task, we require the agent to explore only a small subset of its available action space. This forces the agent to behave in a more focused manner. Interestingly, as the number of allowed actions decreases, the performance of our goal-directed agent tends to improve.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2 Related Work

Vector space models, in which words or groups of words are represented as n-dimensional vectors, have been an active area of research since the 1960s [9]. In recent years, statistical or count-based models have been replaced by embeddings trained from large amounts of uncurated text. In our work, we utilize the skip-thought embedding model presented by Kiros et al. in 2015 [8], which in turn relies on the word2vec Google News word embeddings trained by Mikolov et al. [11]. Skip-thought vectors [8] are an application of the word2vec skip-gram training method at the sentence level, resulting in an encoder that represents each input sentence as a fixed-length vector. (Similar work is also found in the paragraph vectors of Le and Mikolov [10].) Other related work includes [7], who use an encoder-decoder pipeline to learn multimodal embeddings of both images and text. Key weaknesses of these semantic embedding models include susceptibility to triangle inequalities and breakdowns in symmetry [12] as well as their inability to represent different semantic meanings of the same word. Researchers have proposed training methods to achieve word-sense disambiguation [14, 5] as well as methods for combining the strength of multiple embedding algorithms [4].

Of course, an abstract representation of words in terms of other words has only limited use. In order to forge a connection between sentence representations and grounded behaviors in the agent's environment, we build on the affordance-based reasoning methods of Fulda et al. [3]. In some respects, this research resembles the work of Kaplan, Sauer and Sosa in natural language guided gaming [6], in that both architectures enable the agent to improve its performance as a result of text input from a human user. The research differs, however, in the way language understanding is acquired and applied. Rather than learning a task-specific relationship between input and output, our agent leverages the implicit structure of the joint embedding space to prioritize its actions with respect to a human-defined goal.

3 Our Algorithm

Following the precedent established in [3], we model our agent as a purely text-based entity. The agent's state s is a natural language string describing the immediate environment. Actions are represented as verb/noun pairs $a = v + \text{`` ''} + n$ where v is an English-language verb and n is an English-language noun. The agent's objective is to earn points by select action primitives (in this case, a verb and a noun) that (a) function well in combination and (b) progress the agent toward a reward state. Critically, the agent does not incorporate a learning model, nor does it modify its behavior over time. Instead, sensory input, human guidance, and action primitives are jointly represented using a pre-trained skip-thought embedding space [8]. When presented with sensory input, the agent compiles a list of candidate actions which are then filtered based on how well they conform with the human's guidance.

GOAL	VERB	NOUN
unlock door	turn	key
carry water	fill	bottle
wash floor	use	sponge
catch butterfly	swing	net
reach roof	climb	staircase
study geometry	read	textbook
enter castle	cross	drawbridge
enter spaceship	use	airlock
take bath	fill	bathtub
activate machine	flip	switch

Figure 1: Sample canonical analogy set used find goal-driven actions. The full set in our experiments included 17 goal-verb pairs and 18 goal-object pairs.

Our evaluation tasks are drawn from the Autoplay learning environment [15]. These text-based virtual worlds are a challenging and largely unsolved domain: The agent is presented with a text description of its environment such as 'You are standing in an open field next to a white house.' The agent extracts nouns from this text (e.g. 'field', 'house'), and must then determine which of the approximately 30,000 verbs in the human language can be paired with the selected nouns in a way that might produce reward. In general, rewards are obtained when the agent either escapes a restricted area, obtains items, or manipulates items.

Figure 2: Agent architecture. State descriptions and human guidance, encoded as skip-thought vectors, are used to influence final action selections.

GAME	HUMAN-DEFINED GOALS
zork1	'enter house', 'get stuff', 'create light', 'move furniture', 'climb tree', 'unlock locks'
zork2	'enter buildings', 'get stuff', 'create light', 'move things'
zork3	'enter buildings', 'get stuff', 'create light', 'move things'
candy	'get candy', 'search for candy'
omniquest	'get stuff', 'climb tree', 'wear clothing', 'move things', 'dig'
bunny	'enter holes', 'get stuff', 'move things', 'open things', 'burn monsters', 'unlock locks'
detective	'get stuff', 'enter buildings'
mansion	'unlock locks', 'take stuff', 'turn on', 'turn off'
spirit	'open things', 'get scrolls'
zenon	'get stuff', 'look under bed', 'unlock locks', 'turn off light'
cavetrip	'open furniture', 'search furniture', 'get clothes', 'get food', 'get batteries'
parc	'enter buildings', 'get stuff', 'close curtains'

Figure 4: The human-defined goals for each game, reflecting high-level knowledge of the reward structure. Critically, none of the goal texts produce rewards in their own right; instead, they provide guidance that the agent uses to select promising noun-verb combinations.

Our key research question is as follows: can an agent designed to pair nouns and verbs in meaningful ways improve its performance by taking human input into account? In other words, is it possible for the agent to ground its language understanding sufficiently to convert free-form text into a prioritization over possible action primitives?

To address this question, we use a small set of example mappings as shown in Fig 1. Let $\{(g_1,v_1) \ldots (g_i,v_i)\}$ be a set of data points mapping human utterances to verbs that facilitate the human's objective (see Fig 1, first and second columns) and let $(g_1,n_1) \ldots (g_i,n_i)$ be a set of data points mapping human utterances to objects in a similar fashion (Fig 1, first and third columns). Given a state s, human guidance g, and a set of game objects $[o_1, o_2, \ldots o_k]$, our agent prioritizes game objects as follows: we define a canonical guidance-to-noun vector $G_{noun} = \frac{1}{k} \sum_0^k (n_i - g_i)$. We now use a dot product to project each candidate object o_i onto the canonical guidance-to-noun vector G_{noun} as shown in Figure 3. Nouns with the highest dot product are given the most priority.

A canonical goal-to-verb vector is defined similarly: $G_{verb} = \frac{1}{k} \sum_0^k (v_i - g_i)$, and verbs relating to each noun are prioritized in the same manner. If more types of action primitives were needed, additional canonical vectors could be similarly created.

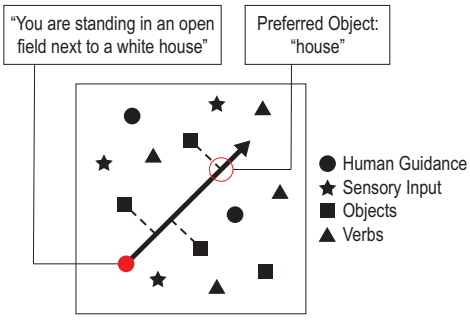

Figure 3: Object prioritization based on human guidance. Candidate objects are projected onto the goal vector G_{noun} and ranked based on magnitude. The system then identifies a set of candidate verbs that are compatible with the chosen object and prioritizes them with respect to G_{verb}.

4 Results

To evaluate our methods, we selected 12 games from the autoplay repository [15]. On each time step, our agent was given one of the human-defined goals depicted in Figure 4. These goals were acquired by allowing a human to examine the early stages of each game and provide a set of behaviors he or she felt were most conducive to point acquisition.

For the most part, these goals were not point-producing actions or even valid action selections within the game. For example, the command 'enter house' in *zork1* produces no change of state. In order to achieve this goal, the agent must navigate to the south side of the house and then execute the commands 'open window' and 'enter window' (which it successfully does). Similarly, when instructed to 'burn monsters' in *bunny*, the agent did not extract the verb 'burn' or the noun 'monsters' from the goal text. Instead, it encoded the goal as a 4800-dimensional vector and from that was able to correctly identify verb/noun pairs that would lead to points. (In this case, the required command was 'burn ooze', and it only works if the player has already executed the command 'get torch'.)

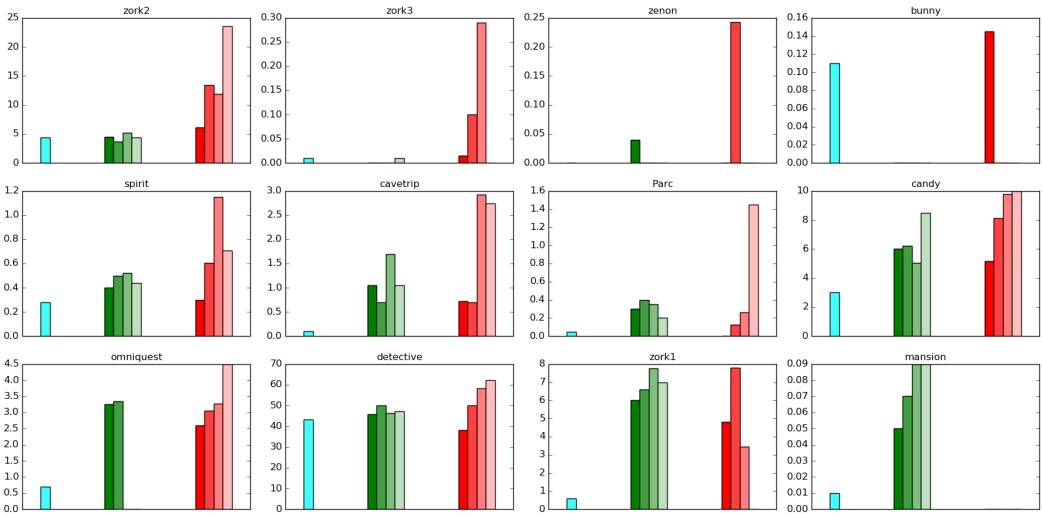

Figure 5: **cyan**: random actions. **green**: no human guidance. **red**: human guidance. Clustered bars show changes in game score as the agent is permitted to try [30,15,3,1] verbs for each object. Agents were allowed to interact with each game for 2000 time steps, and the results from 45 data runs were averaged.

Results are shown in Figure 5. Three types of agent were compared: a naive agent that randomly selected action primitives, an intelligent agent that paired nouns and verbs using the affordance-based method described by [3], and our agent, which combines the affordance-based paring method with human guidance delivered via natural language text. Clustered bars indicate the game score achieved by each agent as environmental constraints were increased: on the leftmost bar, the agent was allowed to attempt 30 different verbs for each noun. Moving right, the number of allowed verbs was restricted to 15, 5, and 1 respectively. The number of allowed nouns was similarly restricted, so that the rightmost red bar indicates the agent's performance when it was allowed to select at most one noun in each state, and was allowed to attempt only its top-priority verb with that noun. Notice that there is a frequent upward trend in the human-guided agent as constraints on the system increase: the agent's ability to convert human guidance into relevant action primitives enabled it to avoid useless and counterproductive behaviors and generate increased reward in a very small number of steps.

5 Conclusion and Future Work

Multimodal embeddings offer exciting new potential for grounding agent observations in physical experience, but simply creating a multimodal embedding space is not enough. We must learn how to navigate that space, and how to leverage the multimodal structure to accomplish new tasks without repeating the training process. This paper presents a method for linking high-level domain knowledge in the form of human-generated text with low-level action primitives. Specifically, we use a 4800-dimensional skip-thought embedding space to encode the agent's inputs, outputs, and guidance offered by a human. In text-based worlds, this approach generates overall improvements in agent performance, and the algorithm is easily generalizable to multimodal embedding spaces. Once a well-structured embedding space has been trained, new prioritizations over behaviors can be created without retraining. A set of twenty or fewer example pairings is sufficient to define the relationship.

In this work we used an embedding space with a single modality (text) to prioritize action primitives. Moving forward we hope to apply our algorithm to multimodal embedding spaces that link text, images, and other sensory input into unified structures that retain semantic meaning. Additionally, these embedding spaces should address the issue of disambiguation between inputs or action patterns that appear in multiple distinct contexts. Stronger multimodal embedding spaces will allow for better inference and average vectors across action modalities. As these embedding spaces become more accurate and reliable, we believe the semantic knowledge found therein will propel autonomous reasoning and natural language understanding to new levels.

References

[1] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017.

[2] Hao Dong, Jingqing Zhang, Douglas McIlwraith, and Yike Guo. I2t2i: Learning text to image synthesis with textual data augmentation. *arXiv preprint arXiv:1703.06676*, 2017.

[3] Nancy Fulda, Daniel Ricks, Ben Murdoch, and David Wingate. What can you do with a rock? affordance extraction via word embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1039–1045, 2017.

[4] Sahar Ghannay, Benoit Favre, Yannick Esteve, and Nathalie Camelin. Word embedding evaluation and combination. In *LREC*, 2016.

[5] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 873–882. Association for Computational Linguistics, 2012.

[6] Russell Kaplan, Christopher Sauer, and Alexander Sosa. Beating atari with natural language guided reinforcement learning. *CoRR*, abs/1704.05539, 2017.

[7] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[8] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. pages 3294–3302, 2015.

[9] Sheldon Klein, Stephen L Lieman, and Gary Lindstrom. Diseminer: a distributional-semantics inference maker. 1966.

[10] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[12] Aida Nematzadeh, Stephan C Meylan, and Thomas L Griffiths. Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words.

[13] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396, 2016.

[14] Joseph Reisinger and Raymond J. Mooney. Multi-prototype vector-space models of word meaning. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2010)*, pages 109–117, 2010.

[15] Daniel Ricks. Autoplay: a learning environment for interactive fiction. `https://github.com/danielricks/autoplay`, 2016.