Learning of Object Properties, Spatial Relations, and Actions for Embodied Agents from Language and Vision

Muhannad Alomari, Paul Duckworth, David C. Hogg and **Anthony G. Cohn**
School of Computing, University of Leeds, UK
(scmara, scpd, d.c.hogg, a.g.cohn)@leeds.ac.uk

Abstract

We present a system that enables embodied agents to learn about different components of the perceived world, such as object properties, spatial relations, and actions. The system learns a semantic representation and the linguistic description of such components by connecting two different sensory inputs: language and vision. The learning is achieved by mapping observed words to extracted visual features from video clips. We evaluate our approach against state-of-the-art supervised and unsupervised systems that each learn from a single modality, and we show that an improvement can be obtained by using both language and vision as inputs.

Introduction

Understanding natural language commands is essential for robotic systems to naturally and effectively interact with humans. In this paper, we discuss our novel loosely-supervised work in acquiring semantic knowledge of natural language commands, given pairs of linguistic and visual inputs. Generally, supervised systems learn from sentences and scenes that have been manually annotated in detail by a human expert. The labelling of data is a labour intensive task that hinders the learning from large corpora, and such labels are not necessarily available for all languages and manipulation scenarios. While unsupervised techniques enable learning from unlabelled data, their performance is usually significantly worse than supervised techniques. In this work, we present a novel multi-sensory loosely-supervised technique capable of acquiring knowledge about object properties, spatial-relations, and actions from unlabelled data by mapping language to vision. Our system learns about objects, relations, and actions from a parallel corpus of n pairs of short video clips $V = \{v_1, \ldots, v_n\}$, and sentences describing these videos $S = \{s_1, \ldots, s_n\}$, as shown in Fig. 1. Our methodology consists of first encoding a number of visual features for each video clip, and utilizes co-occurrences of words and visual features to understand natural language.

Understanding **Natural Language** using multi-sensory inputs has been a long standing objective of AI and cognitive research. Siskind (1996) was one of the earliest researchers to try and understand in a computational setting how children learn their native language and map it to vision. Following his research, in the field of developmental robotics researchers have connected language and vision to teach their robots different concepts; one of the earliest works to do so was a system by Roy *et al.* (1999) which is capable of learning audio-visual associations (i.e. objects' names) using mutual information criteria. Several robotic applications were developed subsequently, such as Steels *et al.* (2001) where language games for autonomous robots are used to teach the meaning of words in a simple static world. Researchers have since developed systems capable of learning objects' names and spatial relations by interacting with a human or robot teacher, as in Steels (2002) and Spranger (2015). The work of Misra *et al.* "Tell me Dave" (2015), and Chai *et al.* "Back to the blocks world" (2014) focused on learning the natural language commands for simple manipulation tasks; this is similar to our work, however we improve on their work in two ways: First, they used a pre-trained natural language parser to extract relevant words from sentences for learning, while we learn from unprocessed linguistic inputs. Second, they assume the robot knows the representations of shapes and spatial relations beforehand, while we extract these by clustering features from video clips.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: Example of "push" action (Sinapov et al. 2016) annotated with the sentence "push the red bottle to the left".

Extracting Visual Clusters

In this section, we describe how we represent the visual input data by extracting a set of visual features from each video clip. Then we show how we abstract values from these features to form a set of clusters. These clusters are used to learn the semantics of words in the following section. We start by processing all video clips to detect and track the objects in each frame. The objects are detected using a table-top object detector (Muja *et al.* 2013), where each object in a video is assigned a unique id, and its location is tracked using a particle filter (Klank et al. 2009).

Next, we obtain three sets of observations: (i) object features: $\{colour, shape, location\}$ of each object; (ii) relational features: $\{direction, distance\}$ of each pair of objects in a scene; and (iii) the *atomic actions* that the robot applies on each object during the video. The features and atomic actions are shown in Fig. 2. The object and relational features are obtained at the initial and final frames of each video, whereas actions are obtained throughout the whole video. It is worth noting that these features are not intended to be exhaustive, but rather to demonstrate our approach; other features could be added as an extension to this work.

Object

- $F_{colour} : object \rightarrow ([0, 360) \times [0, 1] \times [0, 1])$; $F_{colour}(o)$ gives HSV colour values for an object.
- $F_{shape} : object \rightarrow \mathbb{R}^{32}$; $F_{shape}(o)$ gives a 32 bin fast point feature histogram (FPFH) per object.
- $F_{location} : object \rightarrow \mathbb{R} \times \mathbb{R} \times \mathbb{R}$; $F_{location}(o)$ gives an x, y, z location wrt the bottom left corner of the table.

Relational

- $F_{distance} : object \times object \rightarrow \mathbb{R}$; $F_{distance}(o_1,o_2)$ gives the distance between o_1 and o_2 centroids.
- $F_{direction} : object \times object \rightarrow [0, 360) \times [0, 360)$; $F_{direction}(o_1,o_2)$ gives the azimuth and altitude angles from o_1 to o_2.

Atomic Actions

- $F_{action} : object \rightarrow [approach, depart, grasp, discard, lift, lower, move, rotate, null]$; $F_{action}(o)$ gives the atomic action, as shown below.

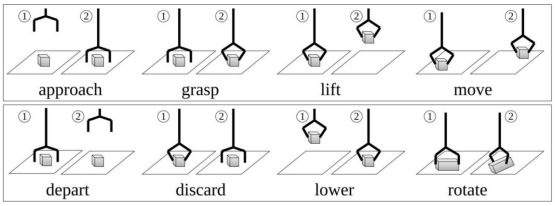

Figure 2: The set of pre-defined features and atomic actions.

Once the observations (objects, relations, actions) have been obtained for all objects in all video clips, we process them to extract unique concepts, e.g. distinguish the shape *cube* from the shape *prism*, and distinguish the colour *red* from the colour *blue*, etc. This is achieved by clustering the values from all observations of each feature space separately to obtain multiple clusters. The extracted clusters are used to construct a visual clusters vector (F) with length equal to the total number of clusters. This forms the list of possible semantic tags for words. For instance, the Dukes (2013) dataset (intended to learn natural language commands) contains four unique shapes: prism, cube, ball, and cylinder. We cluster the shape values of all objects from all video clips and four clusters/shapes emerge $shape_1$ = cube, $shape_2$ = prism, $shape_3$ = ball, and $shape_4$ = cylinder. The same clustering method is used on all observations ($colours$, $locations$, $directions$, $distances$, and $atomic\ actions$) each of which is done separately, i.e. we cluster colours alone, shapes alone, etc. The outputs (or the clusters) are then combined into a single vector $F = \{shape_1, shape_2, shape_3, shape_4, colour_1, \ldots, location_1, \ldots, direction_1, \ldots, distance_1, \ldots, action_1, \ldots\}$ to represent the visual features and is used in the next section to learn words' semantics.

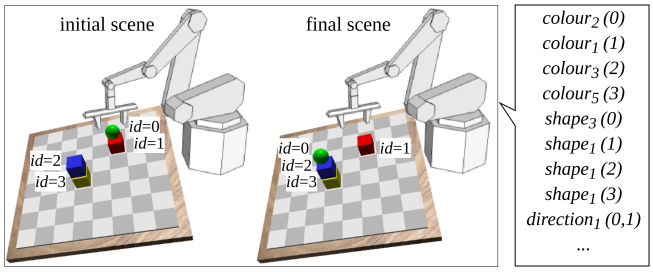

Figure 3: Example "place the green ball above the blue block" (Dukes 2013) represented using the clusters in F.

Learning Visual Semantics

Assigning a semantic category to a word is an essential preprocessing step for understanding and executing natural language commands in robotics. In this section, we show how we connect words to semantic clusters that we extracted in the previous section. The problem statement of this section is: given (1) a corpus of n sentences $S = \{s_1, \ldots, s_n\}$ that contains m unique words $W = \{w_1, \ldots, w_m\}$, and (2) video clips $V = \{v_1, \ldots, v_n\}$ that contain k extracted clusters $F = \{f_1, \ldots, f_k\}$; can we find a partial function Φ, such that it maps words from language to their semantic tags in vision, i.e. $\Phi : W \rightarrow F$. This semantic learning problem is formulated as an assignment problem, where we have to assign words $w_i \in W$ to clusters $f_j \in F$ subject to a cost function $C : W \times F$ that needs to be minimised. We define the cost function as $C_{w,f} = (1 - (N_{w,f}/N_w))$, where $(N_{w,f})$ is the total number of times a word w and a cluster f appear together, and (N_w) is the total number of times the word w appears in the entire dataset. This cost function is equal to zero ($C_{w,f} = 0$), if word w, and feature f, always appear together, and equal to one ($C_{w,f} = 1$) if they are never seen together. This provides a clear indication of whether a word w should be mapped to cluster f or not.

Once the cost function is computed for all word-cluster pairs, we create a cost matrix with words W as rows, and clusters F as columns, as shown in Fig. 4 (left). We then use the Hungarian algorithm (Kuhn 1955) to find the semantic tag for each word by assigning it to its most suitable cluster, as shown in Fig. 4 (right). We also remove function words (such as 'the') by setting a threshold on the Hungarian algorithm. This has the same effect as using term frequency-inverse document frequency (tf-idf) weighting to remove function words (Jones 1972).

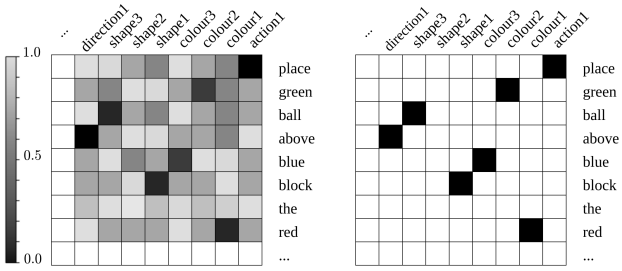

Figure 4: Left: the cost matrix. Right: the output (semantic assignments) using the Hungarian algorithm.

Experimental Procedure

We evaluate the performance of our system using three datasets: a synthetic-world, and two real-world datasets. For the **synthetic-world**, we used the *Extended Train Robots* dataset (Dukes 2013; Alomari et al. 2016) which consists of a thousand short video clips annotated with linguistic commands using Amazon Mechanical Turk, an example is shown in Fig. 3. For the **real-world** datasets, we used (Sinapov et al. 2016; Alomari et al. 2017). In both datasets, a robotic arm is used to manipulate objects placed in front of the robot, and later annotated by a group of volunteers with appropriate natural language commands, examples from both datasets are shown in Fig. 1 and Fig. 5. A summary of all three datasets is presented in Table 1. Note that N/A means a feature space can not be processed for this dataset, e.g. the Sinapov dataset have a single object in each scene, therefore no spatial relations.

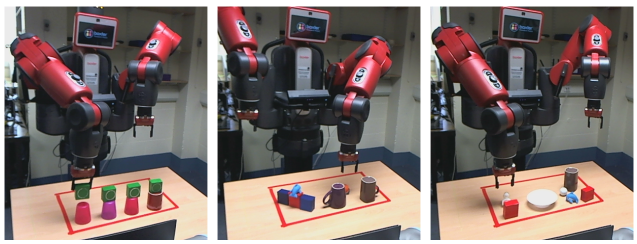

Figure 5: Scenes from Alomari (2017) robotic dataset, where each video is annotated with a natural language command.

Datasets summary							
features	*Col*	*Sha*	*Loc*	*Dir*	*Dis*	*Act*	*Avg*
Alomari	11	13	3	5	2	3	5.3
Sinapov	9	3	N/A	N/A	N/A	6	1
Dukes	9	4	4	3	N/A	4	24.8

Table 1: Number of unique concepts in *colour, shape, location, direction, distance,* and *action* features in all datasets, and the *avarage* number of objects present in each scene.

Implementation Details

The **objects** are detected by applying a tabletop object detector on the first frame in each video, and then tracked throughout the video using a particle filter, as shown in Fig. 6.

During the learning process, we use *atomic actions* (Fig. 2) to represent more **complex actions**. For example a 'pick up' action is represented with (*approach, grasp, lift*) as the robot approaches, grasps and lifts the object; a 'drop' action is represented with just (*discard*) as the robot lets go of the object to fall down on the table. The repetition of such atomic actions across videos forms the action clusters.

We automatically detected **function words** by setting a threshold of $\sigma = .6$ on the Hungarian algorithm, which correctly detects and removes all function words. This translates as a word w being considered a function word if it is not consistent with any cluster $f_j \in F$ for more than 60% throughout the entire dataset.

In our **experiment**, we divided each dataset randomly into four equal folds, to perform four fold cross validation.

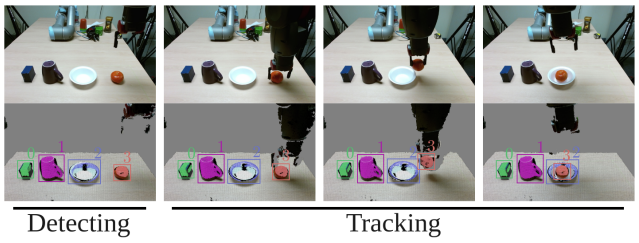

Figure 6: Example of a video sequence "place the orange in the bowl" when the objects are tracked using a particle filter. (Best viewed in colour)

Evaluation

We evaluated the performance of our technique using by its ability to correctly tag words using the learnt semantic tags Φ. To better demonstrate our results in semantic tagging, we compare our technique with (1) a supervised system that learns from labelled data, and with (2) an unsupervised system that learns from unlabelled linguistic data. We consider our **baseline** as the performance of the unsupervised system, i.e. our joint language and vision technique should outperform the unsupervised system that learns from unlabelled linguistic inputs, otherwise there is no benefit of the additional vision component. Similarly, our **upper bound** on performance is the results of the supervised system trained on human labelled (ground-truth) data.

Semantic Tagging Experiment

In this section, we evaluate the system's ability to acquire correct semantic tags for words from parallel pairs of short video clips and linguistic descriptions. The given task is to learn the partial function $\Phi : W \rightarrow F$ that maps words $w_i \in W$ to their corresponding clusters $f_j \in F$, e.g. the word 'red' should be mapped to the cluster colour-*red*.

The results for our semantic tagging experiment are shown in Fig. 7. Here, 'our-system' is compared against (1) the supervised semantic tagger (Fonseca and Rosa 2013) that is trained on human labelled data, and (2) the unsupervised semantic tagger (Biemann 2009) that is trained on unlabelled linguistic data. The results are calculated based on the total number of correct tags assigned to each word in the test fold (four fold cross validation). Note that for the unsupervised system, the results are calculated based on its ability to cluster words that belong to the same category together, i.e. words that describe colours should be given a unique tag different to those that describe shapes, directions, etc. Also, we assign new words in the test fold (words that only exist in the test fold) with a *function word* tag.

Our system is able to correctly learn (85.6%) of the total words in the Dukes dataset, (91.3%) in the Sinapov dataset, and (81.5%) in the Alomari dataset. Compared to only (32.9%, 39.8%, and 31.2% respectively) using the unsupervised system. This clearly shows that adding vision inputs produces more correct and useful semantic tags for words, even though both systems use unlabelled data. A detailed analysis of how the different techniques performed in each feature space is shown in Fig. 8. Note that N/A features from Table 1 have an empty row-column in our analysis.

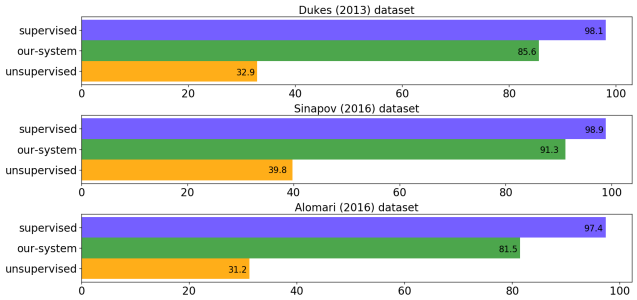

Figure 7: The results of (a) supervised, (b) our approach, and (c) unsupervised semantic learning on all three datasets.

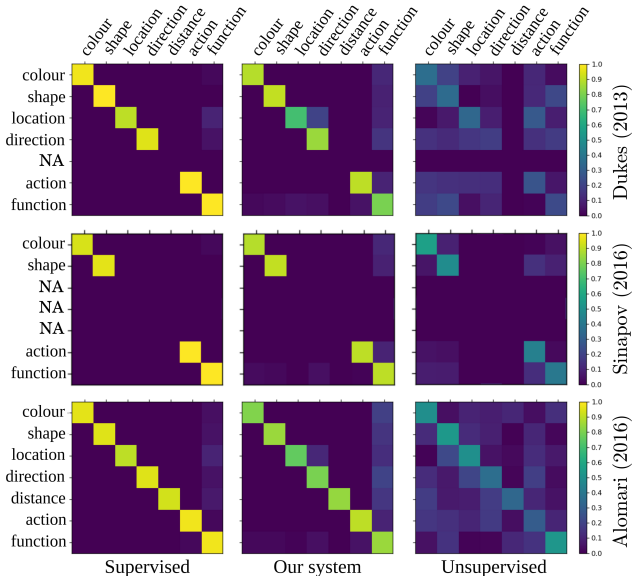

Figure 8: The performance in each feature space for the three systems on all three datasets.

Conclusion and Discussion

We present a system that learns about object properties, spatial-relations, and simple manipulation actions by connecting multi-sensory inputs in a loosely-supervised setting. Our learning framework consists of connecting words from sentences to extracted visual clusters from videos. Our approach outperforms unsupervised semantic tagging techniques and achieves comparable results with supervised systems that learn from labelled data. The approach could be used to ground other modalities such as touch as well.

We use the term *loosely-supervised* to describe the kind of learning that requires the videos and sentences to be temporally aligned beforehand, which is typically used for teaching infants about basic concepts such as colours or shapes. A fully unsupervised system would be able to temporally segment and align long videos and documents and learn from them, which remains an ambition for the future.

Acknowledgements

We thank colleagues in the STRANDS project consortium (http://strands-project.eu) for their valuable comments. We also acknowledge the financial support provided by EU FP7 project 600623 (STRANDS).

References

Alomari, M.; Chinellato, E.; Gatsoulis, Y.; Hogg, D. C.; and Cohn, A. G. 2016. Unsupervised Grounding of Textual Descriptions of Object Features and Actions in Video. In *15th International Conference on Principles of Knowledge Representation and Reasoning*.

Alomari, M.; Duckworth, P.; Hogg, D. C.; and Cohn, A. G. 2017. Natural language acquisition and grounding for embodied robotic systems. In *Proc. AAAI*.

Biemann, C. 2009. Unsupervised part-of-speech tagging in the large. *Research on Language and Computation* 7.

Dukes, K. 2013. Semantic annotation of robotic spatial commands. In *Language and Technology Conference (LTC)*.

Fonseca, E. R., and Rosa, J. L. G. 2013. A two-step convolutional neural network approach for semantic role labeling. In *Neural Networks (IJCNN)*, 1–7. IEEE.

Klank, U.; Pangercic, D.; Rusu, R. B.; and Beetz, M. 2009. Real-time CAD Model Matching for Mobile Manipulation and Grasping. In *9th IEEE-RAS International Conference on Humanoid Robots*, 290–296.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*.

Misra, D. K.; Sung, J.; Lee, K.; and Saxena, A. 2015. Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions. *JAIR* 0278364915602060.

Muja, M., and Ciocarlie, M. 2013. (last accessed 16-12-2016). http://www.ros.org/wiki/tabletop_object_detector.

Roy, D.; Schiele, B.; and Pentland, A. 1999. Learning Audio-Visual Associations using Mutual Information. In *Integration of Speech and Image Understanding, 1999.*

She, L.; Yang, S.; Cheng, Y.; Jia, Y.; Chai, J. Y.; and Xi, N. 2014. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Sinapov, J.; Khante, P.; Svetlik, M.; and Stone, P. 2016. Learning to order objects using haptic and proprioceptive exploratory behaviors. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*.

Siskind, J. M. 1996. A Computational Study of Cross-Situational Techniques for Learning Word-to-Meaning Mappings. *Cognition* 61(1):39–91.

Sparck Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28(1):11–21.

Spranger, M., and Steels, L. 2015. Co-acquisition of syntax and semantics - an investigation in spatial language. In *Proc. IJCAI*.

Steels, L., and Kaplan, F. 2002. Aibo's First Words: The Social Learning of Language and Meaning. *Evolution of Communication* 4(1):3–32.

Steels, L. 2001. Language Games for Autonomous Robots. *Intelligent Systems, IEEE* 16(5):16–22.