Unsupervised Natural Language Acquisition and Grounding to Visual Representations for Robotic Systems

Muhannad Alomari, Paul Duckworth, Yiannis Gatsoulis, David C. Hogg and **Anthony G. Cohn**
Institute for Artificial Intelligence and Biological Systems, University of Leeds, UK
(scmara, scpd, y.gatsoulis, d.c.hogg, a.g.cohn)@leeds.ac.uk

Abstract

We present a cognitively plausible novel framework capable of learning the components of natural language for robotic systems in a real world environment. The system is trained by a "show-and-tell" procedure, in which video clips paired with natural language commands are provided to the system, without any prior knowledge about the mapping between them. The learning approach is an unsupervised technique, that uses a novel relational graph representation to build connections between language and vision. At the same time, a set of probabilistic grammar rules are generated which encode the visual semantics of phrase structure and word classes. We show that the knowledge gained can be used to parse novel linguistic commands involving previously unseen objects.

1 Introduction

Understanding how children learn the components of their mother tongue and the meanings of each word has long fascinated cognitive scientists; equally robots face a similar challenge unless all this knowledge is pre-programmed, which is no easy task either (nor does it solve the problem of language change over time). In this paper we show how a robot can start with no such knowledge and can gradually acquire certain components of language and their groundings in the perceptual world. Researchers have tackled the language acquisition problem using different approaches, such as *individual* and *social learning*. In *individual learning*, the robot is provided with data needed to learn about natural language without any further assistance from the teacher, and is expected to learn from such data in an unsupervised way [Siskind, 1996; Roy *et al.*, 1999; Needham *et al.*, 2005; Alomari *et al.*, 2016]. In *social learning*, the teacher plays an important role in the learning process, by providing feedback to guide the learner in acquiring the different language components [Steels and Kaplan, 2002; Spranger, 2015]. In this research, we follow the *individual approach*, as it enables the learning from large data without the need for constant supervision, thus opening the door to a number of applications such as automatic sports commentators, weather forecasting, and intelligent robotics.

This work aims to answer the following two questions, (i) how can a robot bootstrap its knowledge in language and vision? and (ii) how can it ground language to concepts in vision? To answer these questions in a cognitively plausible setting, we took into consideration that human learning is incremental and it is typically loosely supervised. Also, the system should learn from human description of the world, and, at the same time, the outcome of the learning process should be representable in a form understandable by humans. Keeping the aforementioned in mind, we developed a novel *individual learning* approach capable of acquiring symbolic knowledge in both language and vision simultaneously, and using this knowledge to parse previously unseen natural language commands. The learning is accomplished using a show-and-tell procedure; this is inspired by the fact that children are able to acquire knowledge of their everyday physical world and how to describe it by interacting with their parents in a similar procedure (show-and-tell). Volunteers controlled a robot to perform a variety of table top tasks, which were subsequently annotated with natural language commands, as shown in Figure 1. The recorded videos and commands are used as input data to our system to learn three key components, (i) the words' classes (actions, relations, etc.) in natural language; (ii) the visual representation of these words; and (iii) the grammar rules. To the best of our knowledge, this is the first system that learns all three simultaneously.

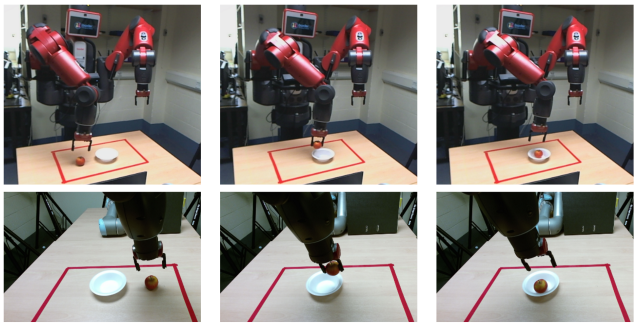

Figure 1: An example of a video sequence for the command "*place the apple in the bowl*". The first video (1^{st} row) shows the entire scene, while the second video (2^{nd} row) shows the robot's view of the scene. The red square on the table is to avoid placing objects outside the robot's view.

2 Related Work

Language acquisition has been a long standing objective of AI and cognitive research. One of the earliest computers capable of understanding natural language commands to perform simple tasks in a virtual world was SHRDLU [Winograd, 1972]. It was pre-equipped with all the linguistic and visual knowledge needed to understand and perform commands such as *pick up the red block*. In this work, we will show how our system can incrementally learn the knowledge needed to perform similar commands in a real-world environment.

In developmental robotics, researchers have combined language and vision to teach their robots about different concepts; one of the earliest works to do so was a system by Roy *et al.* [1999] capable of learning audio-visual associations (basically objects' names) using mutual information criteria. Many more robotic applications were developed subsequently, such as Steels *et al.* [2001; 1995] language games for autonomous robots, used to teach them meaning of words in a simplified static world, or Needham *et al.* [2005] to teach artificial agents to play table-top games. Further, Steels [2002], Spranger [2015], and Bleys [2015] designed systems capable of learning objects' names and certain relations by either interacting with human or robot teachers. Researchers also combined linguistic descriptions from the web with visual features from images to teach their robots different actions, such as setting a table in Dubba *et al.* [2014], or making pancakes in Beetz *et al.* [2011]. Combining language and vision was also used to learn natural language commands for robotic systems, for example, learning linguistic instructions to navigate autonomous mobile robots or drones [Lauria *et al.*, 2002; Huang *et al.*, 2010; Tellex *et al.*, 2011], or performing manipulation tasks [Dukes, 2013; Spranger and Steels, 2015].

In the works discussed above, the learning of concepts was enabled by at least one aspect already being known. In some cases, researchers only presented their systems with a single concept to learn at a time (i.e. a single object in the scene), such that it knows which concept is to be learned. In other cases certain hard-coded knowledge were provided initially (i.e. colours, grammar rules, etc.). In this research, we use a more relaxed set of constraints, and yet we show that our system is still capable of learning about language and vision.

3 Learning Framework

We provide our robot with the ability to learn three components, (i) the words' classes in natural language; (ii) the visual representation of these words; and (iii) the grammar rules. Our learning framework can be summarized by the following steps; (a) the robot receives an RGBD video and its description; (b) each description is represented as a number of tokens (n-grams), and each video as a sequence of graphs that encodes the visual information; (c) these representations are used to build hypotheses that ground n-grams to their visual representations; these hypotheses are tested and used to update the robot's knowledge in language and vision. Further details of (a, b) are given in § 4, and (c) in § 5. We show that a robot can learn about language and vision in a real-world setup in § 6.

4 Knowledge Representation

In this section we describe our representation of the input data: (i) an $RGBD$ video clip, and (ii) a short description.

4.1 Linguistic Input Representation

For each sentence, we aim to match n-grams to their visual representations. To do this, we convert the text to all lower case and remove any punctuation (as this is not explicitly present in spoken language). We then extract all possible n-grams with $n \leq N$. This bag of n-grams is used to match language with visual representations, this is explained in § 5.

4.2 Visual Input Representation

For each video clip, we extract a number of visual representations which we aim to match to language. To do this, we initially detect objects in the video using an unsupervised *table-top_object_detector* package readily available in ROS [Muja and Ciocarlie, 2013] as shown in Fig. 2. Each of these objects is tracked using a particle filter [Klank *et al.*, 2009]. For each object, we compute a set of properties and pairwise relations. We then use these properties and relations to build a sequence of graphs that abstracts the spatio-temporal details of actions.

Figure 2: Clustering the different objects, (a) original scene, (b) point cloud of original scene, (c) table-top algorithm output (object clusters).

Objects Properties and Relations:

For each detected object in the video clip we compute five pre-defined visual features:

1) $colour : object \rightarrow ([0, 360) \times [0, 1] \times [0, 1])^c$; $colour(o)$ gives HSV colour values for object point cloud, where c is the number of points.
2) $shape : object \rightarrow R^{32}$; $shape(o)$ gives a 32 bin fast point feature histogram (FPFH) per object cluster.
3) loc(ation): $object \rightarrow R \times R \times R$; $loc(o)$ gives an x, y, z location of its centroid wrt the robot's base.
4) $dist$(ance): $object \times object \rightarrow R$; $dist(o_1,o_2)$ gives the distance between o_1 and o_2 centroids.
5) dir(ection): $object \times object \rightarrow [0, 360) \times [0, 360)$; $dir(o_1,o_2)$ gives the azimuth and altitude angles from o_1 to o_2.

This set of features is not intended to be exhaustive (but rather to demonstrate the approach); other features could potentially be included. Our robot has no pre-given knowledge in any of these feature spaces, e.g. the number of objects in the world, or the language used to describe them, or any prior discretisation of the feature space. Once the visual features have been computed for all objects in the video, we cluster their values into Gaussian components. The optimal number of components is selected unsupervised using a Bayesian Information Criterion. An example is shown in Fig. 3.

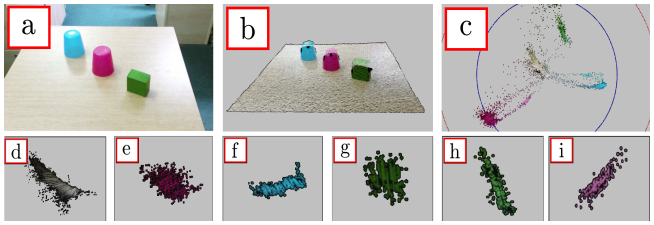

Figure 3: Colours of all pixels in the point-cloud (b) are projected into a single 3D HSV colour space (c), then clustered into separate colours (d - i). The same applies for all features.

Graph Representation

At any given time, we represent the state of the visual world as a directed acyclic graph (DAG) [Christofides, 1975] with nodes that correspond to the visible objects and all relations between pairs of these objects. An ordered pair of arcs connect each relation node to its constituent object nodes. The properties of each object node (colour, location, shape) and relation node (distance, direction) are represented by connected property nodes. The object node corresponding to the *gripper* is distinguished from other objects and has only the location property (Thus the gripper is in a sense special and forms a particular pre-known object type). The value at each property node is either the label of a Gaussian component, when the measurement (e.g. RGBD vector) has Mahalanobis distance within a fixed threshold of this component, or the label 'changing' when outside, to signify that this property is transitioning between components. For our experiments, only the location property of objects changes in this way. An example of our graph representation is shown in Fig. 4.

By omitting consecutive repetitions of identical DAGs, we obtain a sequence of DAGs that represents the video clip. We will refer to each of these DAGs as states, since they describe constant configurations of the visible objects, albeit that some objects may be in motion, denoted by the 'changing' label.

The principle we use for learning is to seek frequent co-occurrences of n-grams of the textual imperatives and sub-graphs or consecutive sequences of sub-graphs extracted from the state sequences derived from the corresponding video clips. The idea is to relate sequences of words to fragments of the visual representation of the world. Ideally we would like to perform the learning on all sequences of all sub-graphs, but this remains an ambition for the future. At present, we steer the learning towards (1) object properties, by extracting all connected sub-graphs involving objects nodes and their properties, (2) relations between objects, by extracting all connected sub-graphs from pair nodes and their properties, and (3) actions, by extracting sequences of sub-graphs that contain the gripper object node, one other object node that has a property with the label 'changing' and the pair node that connects the gripper node with this object node. as shown in Fig. 5. We will refer to these sub-graphs as *graphlets*, where each one has at least one *connection node* (shown in purple) that is used to connect graphlets together. This allows us to reconstruct a graph structure from combination of graphlets.

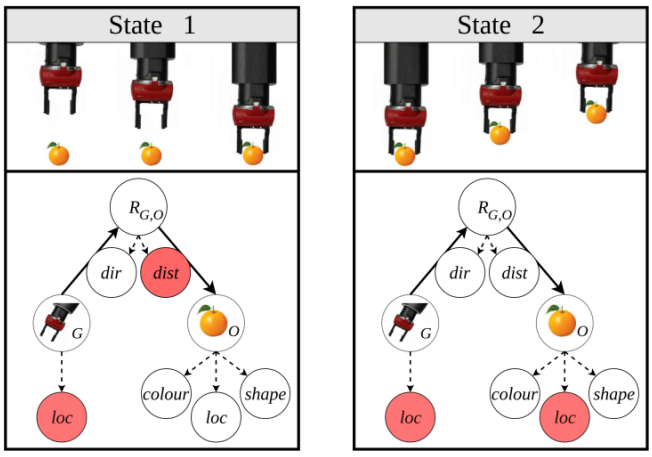

Figure 4: Graph representation for the command "*pick up the orange*" consists of two states. State 1 encodes G(ripper) moving whilst O(range) is static. So the loc(ation) feature node connected to G is 'changing' (red), while the loc node connected to O remains 'constant' (white). State 2 encodes both G and O move together.

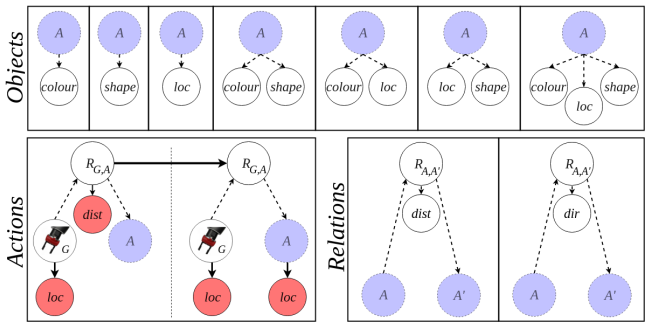

Figure 5: Examples of sub-graphs (graphlets) extracted from the states in Fig. 4.

5 Language Acquisition and Grounding

In this section, we show how we connect words in language (e.g. the 1-gram 'blue') with concepts in vision (e.g. the graphlet representing the colour *blue*), and at the same time build grammar rules that govern the sentence structure.

5.1 Visual representations of Words

In this work, the grounding is achieved using an idea inspired by Hebbian theory. Which can be summarized as: "*Cells that fire together, wire together*" [Schatz, 1992]. This idea is translated to "*n-grams in language and graphlets in vision that appear together, are connected*". As an example, the 1-gram 'blue' and the *blue* colour graphlet will appear consistently together throughout the different videos; therefore should be connected, while the 1-gram 'the' is not consistent with any graphlet; therefore is not connected to any graphlet (this is how the robot comes to know that 'the' is a functional word). To measure the consistency between n-grams and graphlets, we follow the frequentist approach. We keep track of the number of times an n-gram and a graphlet appear

individually, and the number of times the two appear together. We use these frequencies to compute the conditional probabilities that associate each n-gram with a graphlet using:

$$P(g|n) = \frac{F_{gn}}{F_n}, \quad (1)$$

where n is an n-gram, g is a visual graphlet, F_n is the frequency at which n appeared individually, F_{gn} is the frequency of seeing both n and g together. This probability function is computed between every n-gram and graphlet. We filter out the unlikely associations by keeping only the maximum likelihood values for every n-gram in every visual feature space. The robot ends up with a number of associations that need to be verified; we will show how the verification process is achieved in the following section.

5.2 Validation of Associations

Once strong associations have been generated (between n-grams and graphlets), we attempt to validate them by using previously seen videos and sentences. For example, the 1-gram 'blue' might have a high likely association with two different graphlets, one representing the colour feature *blue*, and the other representing (something incorrect) the shape feature *cube*. This can occur due to noise or insufficient data.

The validation occurs by examining how these associations compare with the previously seen videos and sentence pairs. We start this process by first translating the input sentence into multiple graphs. This is done by first representing all n-grams in the sentence with their highly associated graphlets; and then create multiple graph structures by connecting the graphlets together in various orders (the order is important and will later map to learning grammar). We call these graphs *hypothesis graphs*. Each hypothesis graph (from a sentence) is compared against its corresponding input video graph sequence, and if any match (i.e. the hypothesis graph is an induced sub-graph of the input graph), then we have validated the associations for these n-grams. For example, consider the sentence given in Fig. 4: "*pick up the orange*", and suppose that our robot does not know the meanings of any of the words in this sentence. However, it associates the 2-gram 'pick up' with one action graphlet, and the 1-gram 'orange' with two possible object graphlets, whilst 'the' has no strong associations. These n-grams and graphlets are shown in Fig. 6. To validate these, multiple hypotheses graphs are generated that reflect all possible combinations. This is done by connecting the *connection nodes* (shown in purple) in both the action graphlet and the object graphlets together, shown in Fig. 6 (A, B). We then check which (if any) of the generated hypotheses graphs match the input video. Since the hypothesis graph shown in Fig. 6-(A) matches with the input video graph shown in Fig. 4, the robot has validated the associations used to build this graph and correctly grounded the n-grams 'pick up' and 'orange' with their visual graphlets.

5.3 Learning Grammar Rules

In order to understand linguistic commands, the robot needs to learn grammar rules that govern sentence structure. To highlight this, consider the example command "*place the orange in the bowl*". Even assuming the robot has a correct visual representation (graphlet) for each word, shown in Fig. 7, it still needs an understanding of which object should be placed where. This translates to knowing that the action graphlet 'place' changes the location of the 'orange' object and not the 'bowl' object, and further, that it needs to change the orange's location to a final value described by the relation-object graphlets 'in the bowl'.

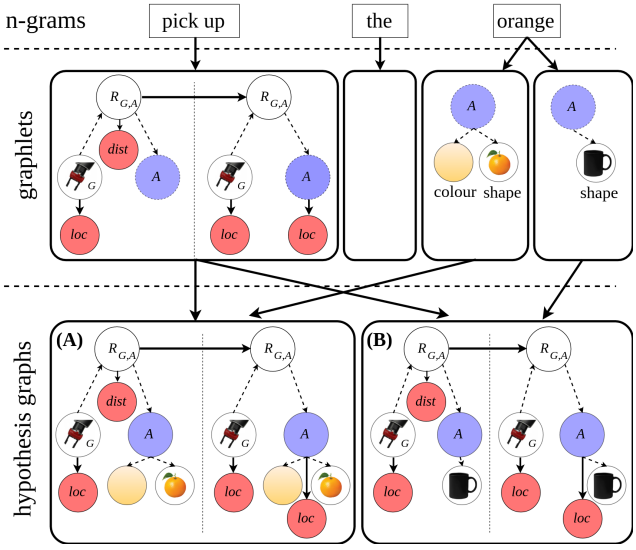

Figure 6: n-grams and graphlets associations, which generates *hypotheses graphs*.

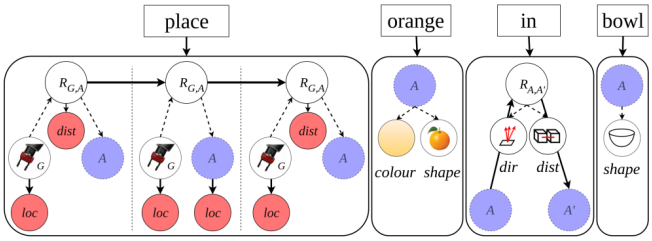

Figure 7: The associated graphlets of the n-grams 'place', 'orange', 'in' and 'bowl', 'the' has no associated graphlets.

To acquire such knowledge, we use the correctly matched hypothesis graphs (previously described in § 5.2) and generate their *syntactic trees*. These trees describe how n-grams should be connected (ordered) in the input sentence, based upon how their corresponding graphlets are connected in the matched hypothesis graph. By using only the correctly matched hypotheses graphs we follow Chomsky's Universal Grammar theory [1965], which states that humans are born with a set of constraints that are hard wired into their brains, and which they use to organise language. As an example, for the command shown in Fig. 4: "*pick up the orange*", the correct hypothesis graph (graph (A) in Fig. 6) is used to generate its equivalent syntactic tree. The matched hypothesis graph encodes the knowledge of which objects are manipulated by the actions in the input sentence. This information is

mapped into a syntactic tree, as shown in Fig. 8. The n-grams in the input sentence are semantically mapped to their corresponding graphlet types (e.g. 'pick up' is an *Action graphlet*, therefore is labelled *Action*) The non-terminals (e.g. *Action*, *Object*) are called this way for readability, the robot does not know these names specifically, though it does know that these several different kinds of knowledge exist and correspond to different parts of the visual representation.

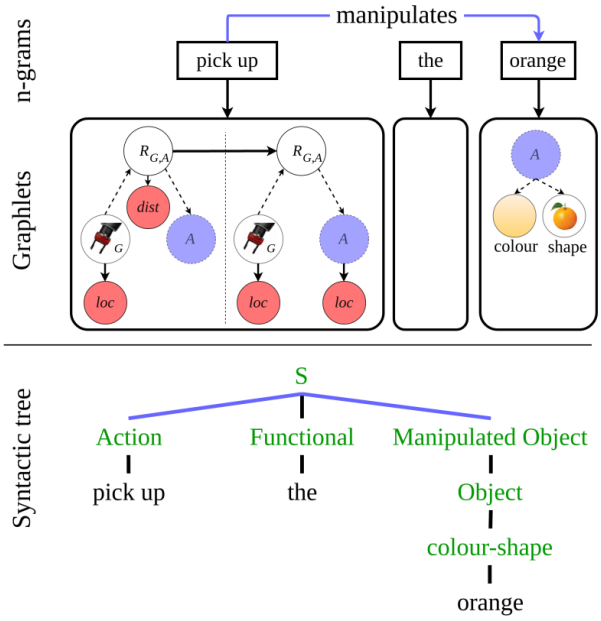

Figure 8: Example of a syntactic tree generated from the correctly matched hypothesis graph.

To learn the grammar rules from syntactic trees, we initiate our robot with an empty Probabilistic Context Free Grammar (PCFG) rule set. The PCFG models each grammatical rule by assigning it a probability, where the probability of each rule is proportional to the number of times the robot observes. This idea agrees with the findings of Hudson-Kam and Newport [2005], which shows that children reproduce the most frequent grammatical forms they hear. Grammar rules learned from only this example are shown in Table. 1, however, rules from all input examples are accumulated into one set.

Learning Grammar Rules	
Grammar Rules	*Probability*
Action → *pick up*	1.0
Functional → *the*	1.0
colour-shape → *orange*	1.0
S → *Action, Functional, M-Object*	1.0
Manipulated Object → *Object*	1.0
Object → *colour-shape*	1.0

Table 1: Learning grammar rules from the syntactic tree shown in Fig. 8

6 Experimental Validation and Dataset

We evaluate the performance of our system using two datasets, a simple real-world setup and a synthetic dataset.

For the **real-world** setup, we used a Baxter robot as our test platform and attached a Microsoft Kinect2 sensor to its chest, as shown in Figure 1. This was used to collect RGBD videos of Baxter performing various manipulative tasks with real objects from the robot's point of view. We collected a dataset consisting of 160 videos in which volunteers controlled Baxter robot's arms, and manipulated real objects. These objects were tracked and their features extracted as described above. The videos were then annotated with appropriate natural language commands (by a separate group of volunteers). This dataset contains a total of 984 commands (average of six-per video). A variety of different objects were manipulated during the videos such as basic block shapes, fruits, cutlery, and even office supplies. The aim is that our system will match the n-grams used to describe these objects to their correct visual graphlets. A further 40 new videos along with 40 new commands were collected and used as a test set which include new objects which were not present in the training set.

For the **synthetic world**, we used the *Train Robots* dataset (http://doi.org/10.5518/32) which was designed to develop systems capable of understanding verbal spatial commands described in a natural way [Dukes, 2013]. Non-expert users were asked to annotate appropriate commands to 1000 pairs of different scenes. Each scene pair is represented by an initial and desired goal configuration; we automatically animated these to produce videos. 7752 commands were collected using Amazon Mechanical Turk describing the 1000 scenes. We also translated all the commands from English to Arabic, particular care was taken on not to alter any command or change any mistakes in any of them. An example of the dataset if shown in Fig. 9.

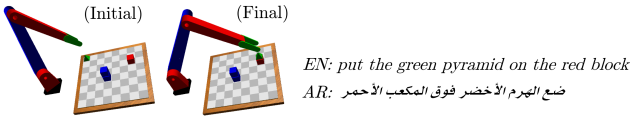

Figure 9: An Example from the *Train Robots* dataset, the Arabic sentence is translated from the English one.

6.1 Evaluation

We evaluated the performance of our system using two measures: (i) its ability to correctly ground n-grams to visual graphlets and therefore learn the groundings of words; and (ii) its ability to correctly parse previously unseen commands.

Grounding n-grams to graphlets

In this section, we evaluate the system's ability to acquire the correct visual-linguistic groundings given the training data. The system's task is to learn the different words associated with each feature space, i.e. that the word 'red' matches to a graphlet containing the colour feature node with a Gaussian component representing a HSV value of red. We define a correct matching by manually inspecting the matched n-gram-

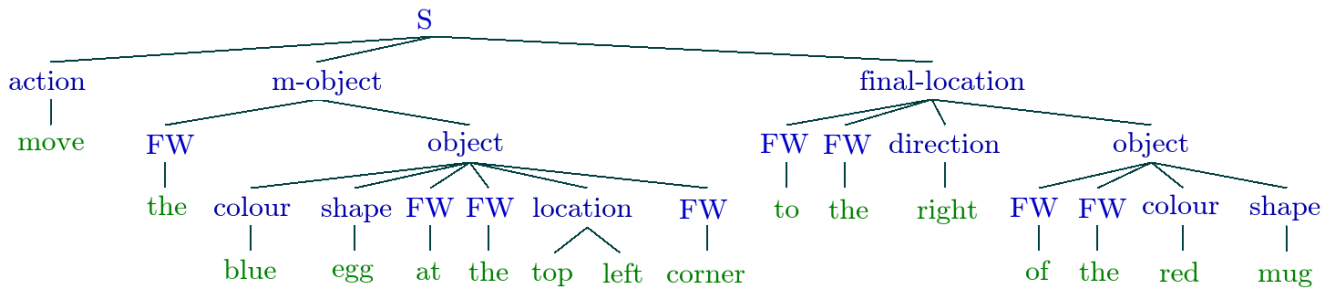

Figure 10: The syntactic tree generated for the new command "*move the blue egg at the top left corner to the right of the red mug*" using the knowledge gained from the training videos.

graphlet pairs and checking if the Gaussian component falls within a range the n-gram describes, e.g. red $\approx$ HSV$(0, 1, 1)$.

Our system was able to correctly ground $47/53$ (88.6%) n-grams to their visual graphlets in the real-world dataset, and $72/81$ (88.9%) in English and $90/101$ (89.1%) in Arabic in the synthetic dataset. A detailed analysis of how the system performed in learning concepts in each feature space is shown in Table 2. Below is a list of examples of the learnt n-grams which were used in the linguistic commands:

1) Colours: *red; yellow; green; blue; pink; black; purple.*
2) Shapes: *block; mug; ball; banana; dolphin; duck; can.*
3) Locations: *top centre; centre; middle; top right; top left.*
4) Directions: *right; left; behind; under; inside; top.*
5) Distances: *far; near; close to.*
6) Actions: *pick up; put down; place; move; pile; shift.*

The system couldn't learn the visual representation of all n-grams due to noise or lack of training data. For example, the n-gram *cyan* was mentioned only once in the real-world dataset and therefore the system did not manage to correctly associate it with its matching colour graphlet.

Grounding n-grams results			
Features	*Real-world*	*Synthetic-English*	*Synthetic-Arabic*
Colours	12/14	15/16	30/31
Shapes	16/18	18/18	22/24
Location	6/7	17/17	16/16
Direction	6/7	10/10	10/10
Distance	3/4	*N/A*	*N/A*
Actions	4/4	12/20	12/20
Total	47/53	72/81	90/101

Table 2: Results of learning the n-grams visual representations from two different datasets.

The use of Gaussian Components to represent visual features allows for efficient and incremental learning, the robot uses an incremental Gaussian Mixture Model approach [Song and Wang, 2005] to update the different graphlets, which allowed us to represent all the input videos without the need to store the entire data. All the results presented in this section are computed using n-grams of $n \leq 3$.

Parsing Novel Commands

We also evaluate the system's ability to generalise its acquired knowledge to new objects and to parse novel commands. This is done using the set of learnt grammar rules.

In the real world dataset, a total of 139 grammar rules were acquired from the 160 training videos. Which we used to test 40 previously unseen videos and commands. In 35 (87.5%) of the test videos the system was able to translate the command into a fully correct syntactic tree. An example of a new command "*move the blue egg at the top left corner to the right of the red mug*" and its generated syntactic tree are shown in Fig. 10. The objects "*red mug*" and "*blue egg*" were not shown to the system in any of the training videos. A sample of the acquired grammar rules that were used in parsing this command is presented in Table. 3.

In the synthetic dataset, a total of 533 grammar rules in the English, and 1344 in the Arabic language were acquired. Which we used to test on 1343 commands. In 929 (69.2%) of these commands the system was able to translate the command into a fully correct syntactic tree. Which is higher than the state-of-the-art supervised system, standing at (60.9%).

7 Conclusion and Future Work

We have demonstrated for the first time in a developmentally plausible setting, that a system can simultaneously learn three kinds of knowledge in an unsupervised manner for processing language and vision from real-world and synthetic data: (i) the words' classes (verbs, relations, objects properties) in natural language; (ii) the visual representation of these words; and (iii) the grammar rules. The learning of grammar rules took inspiration from both universal and probabilistic grammar theories. The 2-level graph representation is also a key contribution of the paper acting as an intermediary representation between the continuous perceptual space, and the purely symbolic linguistic structures. We plan to extend our system to learn language generation from video clips using the gained knowledge.

8 Acknowledgments

We thank colleagues in the School of Computing Robotics lab and 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).

Learnt Grammar Rules	
Non-Terminals	
S → action, m-object, final-location	0.857
m-object → FW, object	0.588
final-location → FW, FW, direction, object	0.058
object → colour, shape, FW, FW, location, FW	0.048
object → FW, FW, colour, shape	0.16
location → top, left	0.324
Non-Terminal Leafs	
colour → *blue*	0.126
colour → *red*	0.264
shape → *mug*	0.14
shape → *egg*	0.04
action → *move*	0.422
direction → *right*	0.404
FW → *the*	0.311
FW → *of*	0.162
FW → *to*	0.123
FW → *at*	0.076
FW → *corner*	0.008

Table 3: The grammar rules used to parse the command shown in Fig. 10 (FW stands for Functional Word, which is a word that has no representation in our pre-defined visual feature spaces).

References

[Alomari *et al.*, 2016] Muhannad Alomari, Eris Chinellato, Yiannis Gatsoulis, David C. Hogg, and Anthony G. Cohn. Unsupervised Grounding of Textual Descriptions of Object Features and Actions in Video. In *15th International Conference on Principles of Knowledge Representation and Reasoning*, 2016.

[Beetz *et al.*, 2011] Michael Beetz, Ulrich Klank, Ingo Kresse, Andres Maldonado, Lorenz Mosenlechner, Dejan Pangercic, Thomas Ruhr, and Moritz Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 529–536. IEEE, 2011.

[Chomsky, 1965] Noam Chomsky. Aspects of the theory of syntax. *Multilingual Matters: MIT Press*, 1965.

[Christofides, 1975] Nicos Christofides. *Graph Theory An Algorithmic Approach*. New York: Academic Press Inc, 1975.

[Dubba *et al.*, 2014] Krishna SR Dubba, Miguel R De Oliveira, Gi Hyun Lim, Hamidreza Kasaei, Luıs Seabra Lopes, Ana Tomé, and Anthony G Cohn. Grounding Language in Perception for Scene Conceptualization in Autonomous Robots. In *Qualitative Representations for Robots: Papers from the AAAI Spring Symposium. Technical report*, pages 26–33, 2014.

[Dukes, 2013] Kais Dukes. Train Robots: A Dataset for Natural Language Human-Robot Spatial Interaction through Verbal Commands. In *International Conference on Social Robotics (ICSR). Embodied Communication of Goals and Intentions Workshop*, 2013.

[Huang *et al.*, 2010] Albert S Huang, Stefanie Tellex, Abraham Bachrach, Thomas Kollar, Deb Roy, and Nicholas Roy. Natural language command of an autonomous micro-air vehicle. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2663–2669. IEEE, 2010.

[Hudson Kam and Newport, 2005] Carla L Hudson Kam and Elissa L Newport. Regularizing unpredictable variation: The roles of adult and child learners in language formation and change. *Language Learning and Development*, 1(2):151–195, 2005.

[Klank *et al.*, 2009] Ulrich Klank, Dejan Pangercic, Radu Bogdan Rusu, and Michael Beetz. Real-time CAD Model Matching for Mobile Manipulation and Grasping. In *9th IEEE-RAS International Conference on Humanoid Robots*, pages 290–296, Paris, France, December 7-10 2009.

[Lauria *et al.*, 2002] Stanislao Lauria, Guido Bugmann, Theocharis Kyriacou, and Ewan Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38(3):171–181, 2002.

[Muja and Ciocarlie, 2013] Marius Muja and Matei Ciocarlie. tabletop object detector - ROS Wiki. `http://www.ros.org/wiki/tabletopobjectdetector`, 2013.

[Needham *et al.*, 2005] Chris J Needham, Paulo E Santos, Derek R Magee, Vincent Devin, David C Hogg, and Anthony G Cohn. Protocols from perceptual observations. *Artificial Intelligence*, 167(1):103–136, 2005.

[Parde *et al.*, 2015] Natalie Parde, Adam Hair, Michalis Papakostas, Konstantinos Tsiakas, Maria Dagioglou, Vangelis Karkaletsis, and Rodney D Nielsen. Grounding the Meaning of Words through Vision and Interactive Gameplay. *Proceedings IJCAI 2015*, 2015.

[Roy *et al.*, 1999] Deb Roy, Bernt Schiele, and Alex Pentland. Learning Audio-Visual Associations using Mutual Information. In *Integration of Speech and Image Understanding, 1999. Proceedings*, pages 147–163. IEEE, 1999.

[Schatz, 1992] Carla J Schatz. The Developing Brain. *Scientific American*, 267(3):60–67, 1992.

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

[Song and Wang, 2005] Mingzhou Song and Hongbin Wang. Highly Efficient Incremental Estimation of Gaussian Mixture Models for Online Data Stream Clustering. In *Defense and Security*, pages 174–183. International Society for Optics and Photonics, 2005.

[Spranger and Steels, 2015] Michael Spranger and Luc Steels. Co-acquisition of syntax and semantics - an

investigation in spatial language. In Qiang Yang and Michael Wooldridge, editors, *IJCAI'15: Proceedings of the 24th international joint conference on Artificial intelligence*, pages 1909–1905. AAAI Press, Palo Alto, US, 2015.

[Spranger, 2015] Michael Spranger. Incremental Grounded Language Learning in Robot-Robot Interactions - Examples from Spatial Language. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2015 Joint IEEE International Conferences on*, pages 196–201. 2015.

[Steels and Brooks, 1995] Luc Steels and Rodney Brooks. *The artificial life route to artificial intelligence: Building embodied, situated agents*. L. Erlbaum Associates Inc., 1995.

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

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

[Tellex *et al.*, 2011] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32(4):64–76, 2011.

[Winograd, 1972] Terry Winograd. Understanding natural language. *Cognitive Psychology*, 3(1):1–191, 1972.