Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Natural Language Acquisition and Grounding for Embodied Robotic Systems

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

Abstract

We present a cognitively plausible novel framework capable of learning the grounding in visual semantics and the grammar of natural language commands given to a robot in a table top environment. The input to the system consists of video clips of a manually controlled robot arm, paired with natural language commands describing the action. No prior knowledge is assumed about the meaning of words, or the structure of the language, except that there are different classes of words (corresponding to observable actions, spatial relations, and objects and their observable properties). The learning process automatically clusters the continuous perceptual spaces into concepts corresponding to linguistic input. A novel relational graph representation is used to build connections between language and vision. As well as the grounding of language to perception, the system also induces a set of probabilistic grammar rules. The knowledge learned is used to parse new commands involving previously unseen objects.

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 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 to learn about natural language without any further assistance from a teacher, and is expected to learn from such data (Siskind 1996; Roy, Schiele, and Pentland 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 different language components (Steels and Kaplan 2002; Spranger 2015). In this research, we follow the *individual approach*, as it enables learning from large datasets without the need for constant supervision, such as learning from *youtube* videos.

This work aims to answer the following two questions, (i) can a robot bootstrap its knowledge in language and vision? and (ii) can it ground language to concepts in vision? To answer these questions in a cognitively plausible setting, we take into consideration human learning which is incremental and is typically loosely supervised. Further, our system is tasked with learning incrementally from human description of the world, while the outcome of the learning process should be representable in a form also understandable by humans. We present a novel *individual learning* approach capable of acquiring symbolic knowledge in both language and vision simultaneously, and use this knowledge to parse previously unseen natural language commands. The learning is accomplished using a show-and-tell procedure; this is inspired by how children acquire knowledge of their everyday physical world by interacting with their parents. 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 two key components, (i) the words' visual representation (actions, spatial-relations, and object properties); and (ii) the grammar rules. To the best of our knowledge, this is the first system that learns these concurrently.

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 more suitable for teaching infants about basic concepts such as colours or shapes. A fully unsupervised system would be able to learn from longer non-segmented videos and documents (i.e. be able to temporally segment and align long videos and documents), which remains an ambition for the future.

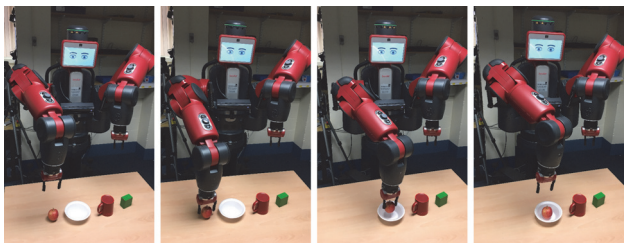


Figure 1: An example of a video clip annotated with the natural language command "*place the apple in the bowl*".

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, our system can incrementally acquire the knowledge needed to perform similar commands from a real-world table-top environment.

Understanding how children learn the language and the meanings of each word has long fascinated cognitive scientists. 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 was capable of learning audio-visual associations (i.e. objects' names) using mutual information criteria. Other robotic applications were developed subsequently, such as the robotic language games by Steels *et al.* (1995; 2001) used to teach robots meaning of words in a simple static world, or Needham *et al.* (2005) to teach artificial agents table-top games. Further, Steels (2002), Spranger (2015), and Bleys (2015) designed systems capable of learning objects' names and spatial relations by 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 is also used to learn natural language commands for robotic systems, for example, learning linguistic instructions to navigate autonomous mobile robots (Lauria et al. 2002; Huang et al. 2010; Matuszek et al. 2013), or to navigate autonomous drones (Tellex et al. 2011), or perform manipulation tasks. 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, which is similar to our work; however, we improve on their work in a number of ways. For instance, they assumed the shapes, spatial-relations and actions representations are known to the robot beforehand, while we extract these automatically by clustering features from video clips. Also, they rely on a manually constructed parser to extract the verbs and nouns from natural language, while we learn these grammar rules from the raw natural language data.

Learning Framework

We aim to learn two components, (i) words' visual representation in videos (actions, spatial relations, object properties); and (ii) the language grammar rules. Our framework, shown in Fig. 2, can be described by the following steps;

(a) **Input:** the robot starts with a short temporally aligned video clip and an annotated natural language description. We temporally segment the videos/sentences such that each segment contains a single action/verb in it.

(b) **KR:** each linguistic description is represented as a sequence of tokens (words), and each video as a sequence of spatio-temporal directed acyclic graphs (STDAG) (Christofides 1975) that encodes the visual information. Further details are given in §Knowledge Representation

(c) **Language Acquisition:** the two representations (graphs, words) are mapped together to build hypotheses that bootstrap the robot's knowledge in language and vision. Further details are given in §Language Acquisition.

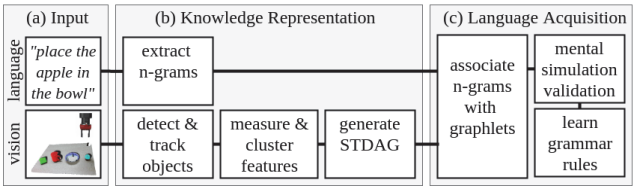


Figure 2: The learning framework.

Knowledge Representation

In this section we describe our representation for the input language and vision data. We process each video clip and its linguistic description separately.

Linguistic Input Representation: For each sentence, we aim to match sequences of words (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 $n \leq N$. The extracted n-grams are used to map natural words to vision, details in §Language Acquisition.

Visual Input Representation: For each video clip, we encode a number of visual representations which we aim to match to language. To do this, we initially detect objects in the video using a table-top object detector (Muja and Ciocarlie 2013). Each of these objects is tracked using a particle filter (Klank et al. 2009). For each detected/tracked object, we obtain measurements of its properties and spatial pairwise relations with other objects at every frame of the video clip. The measurements include (*colour, shape, location, relative direction, relative distance*). This set of features is not intended to be exhaustive (but rather to demonstrate the approach); other features could be included. The features (measurements) are presented below:

- $F_{colour} : object \rightarrow ([0, 360) \times [0, 1] \times [0, 1])$; $F_{colour}(o)$ gives HSV colour values for an object.
- $F_{shape} : object \rightarrow \mathbb{R}^{32}$; $F_{shape}(o)$ gives a 32 bin rotation and translation invariant shape fast point feature histogram (FPFH) per object (Rusu 2009).
- $F_{location} : object \rightarrow \mathbb{R} \times \mathbb{R} \times \mathbb{R}$; $F_{location}(o)$ gives an x, y, z location wrt the bottom left corner of the table.
- $F_{distance} : object \times object \rightarrow \mathbb{R}$; $F_{distance}(o_1,o_2)$ 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.

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 spaces. Once the measurements have been obtained for all objects, we cluster their values into Gaussian components. Each feature space is clustered separately, i.e. we cluster colours alone, shapes alone, etc. The optimal number of components is selected unsupervised using a Bayesian Information Criterion (BIC). These Gaussian components are used as concepts that facilitate the grounding of natural language to vision, e.g. the word '*red*' is grounded to the extracted/clustered *red* colour Gaussian component in the HSV colour feature space. An example for detecting and tracking objects, and for obtaining measurements and clustering the colour feature space is shown in Fig. 3. In our incremental learning process, the robot is introduced to new concepts over time (e.g. new colours, distances, etc). When this happens, new concepts that are seen for the first time should be created, and the ones that have been seen before should be updated. For example, in our system the red colour boundaries in the HSV feature space is not defined beforehand, and the colour *red* may appear as two different Gaussian components in two different videos, due to lighting or different shades of red in different objects. To address this issue, we use an Incremental Gaussian Mixture Model (IGMM) approach (Song and Wang 2005) to merge or create new concepts. The IGMM works in two steps: (i) decide whether an observed Gaussian component has been seen before using statistical tests, (ii) update the existing component if it has been seen before, otherwise create a new component in the feature space. After clustering and updating the measurements values, we use the clusters (Gaussian components) to build a sequence of STDAGs that abstracts the spatio-temporal details of actions that occur in the video.

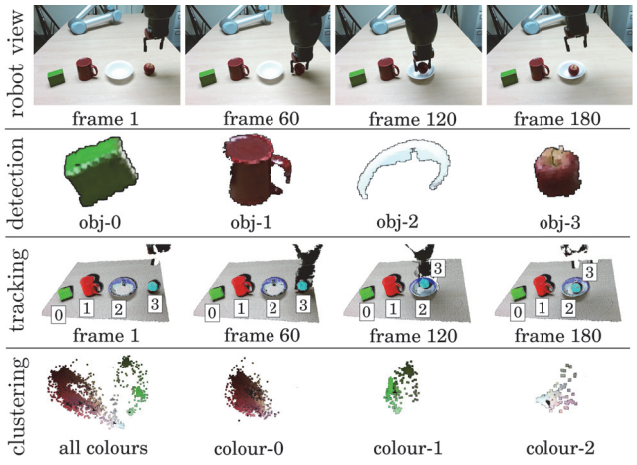


Figure 3: Processing the example in Fig. 1. 1^{st} row shows the robot's view of what happened in the scene. 2^{nd} row shows the detected objects using table-top algorithm. 3^{rd} row shows the tracking results using particle filter. 4^{th} row shows the *HSV* colour measurements of the detected objects in this video (all colours),and the clustering results (Gaussian components colour-0,1,2) shows the unique colours.

Graph Generation: Spatio-temporal graphs have been used in the literature to learn activity models (Sridhar *et al.* 2010, Duckworth *et al.* 2016; 2017, Gatsoulis *et al.* 2016). Our graphs are an extension to their work as we encode more object features. At any given time in a video clip, we represent the state of the visual world as a spatio-temporal directed acyclic graph (STDAG) with nodes that correspond to the visible objects and all relations between pairs of these objects. We encode one node for each object in the video along with connected feature nodes (i.e. *colour*, *location*, *shape*); and one relational node for each pair of objects along with connected pairwise relation nodes (i.e. *direction*, *distance*). An ordered pair of edges connect each relation node to its constituent object nodes. The object node corresponding to the *gripper* is distinguished and is special as it is assumed to form a particular pre-known object type that has only the *location* feature node connected to it. 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. An example of our graph representation is shown in Fig. 4. By omitting consecutive repetitions of identical STDAGs (i.e. consecutive frame that has the same state of the world in it), we obtain a sequence of unique STDAGs that represents the video clip. We will refer to each of these STDAGs as states, since they describe constant configurations of the visible objects, albeit that some objects may be in motion denoted by the 'changing' label.

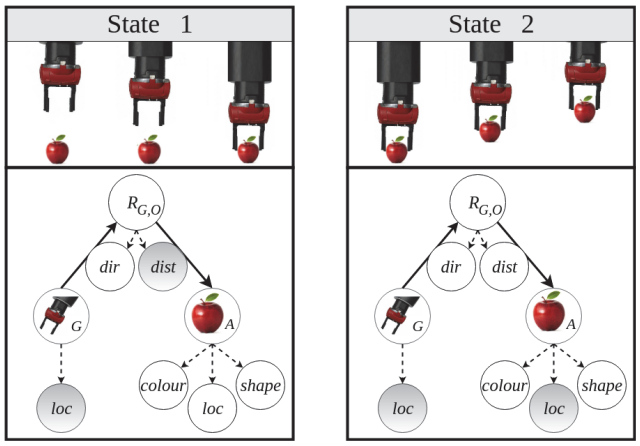


Figure 4: Graph representation for the command "*pick up the apple*" consists of two states. State 1 encodes G(ripper) moving whilst A(pple) is static. So the *location* feature node connected to G is 'changing' (gray), while the *location* node connected to A remains 'constant' (white). State 2 encodes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

The principle we use for learning is to seek frequent co-occurrences of n-grams 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 n-grams 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. We will refer to these sub-graphs as *graphlets* as shown in Fig. 5, Where each graphlet has at least one *connection node* (denoted with c) that is used to connect graphlets together. This allows us to reconstruct a graph structure from combination of graphlets.

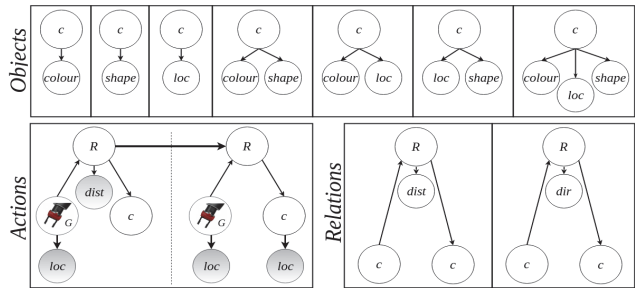


Figure 5: Examples of sub-graphs (graphlets) extracted from the states in Fig. 4.

Language Acquisition and Grounding

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

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 '*apple*' and the *apple* shape 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 function 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:

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

where n is an n-gram, g is a 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. By using n-grams, we end up with a number of n-grams that map to the same visual concept, some of which are incorrect. E.g. the n-grams ('*red*', '*the red*', and '*the red apple*') are all connected to the same *red* colour graphlet with high probability; we need to filter out the incorrect ones '*the red*' and '*the red apple*'. This is achieved by case analysis, consider the case of whether to accept an n-gram ab, consisting of smaller n-grams a and b (e.g. '*the red*' consists of '*the*' and '*red*'). Let g_{ab}, g_a, g_b be the graphlets with the highest association to the n-grams ab, a and b. There are four possible cases shown by the rules below from which we can figure out which ones of these n-grams is incorrect. The accepted hypotheses are shown on the right.

$$g_{ab} = g_a = g_b \quad \rightarrow \quad g_{ab} \tag{2}$$
$$g_{ab} = g_a \neq g_b \quad \rightarrow \quad g_a, g_b \tag{3}$$
$$g_{ab} = g_b \neq g_a \quad \rightarrow \quad g_a, g_b \tag{4}$$
$$g_a \neq g_{ab} \neq g_b \quad \rightarrow \quad g_{ab}, g_a, g_b \tag{5}$$

Rule (2) filters out the smaller incorrect n-grams, by allowing complex n-grams to subsume their constituent ones when all of them are equal. The intuition behind it can be seen in examples like the n-grams '*pick up*', '*pick*' and '*up*' where we want to keep the longer n-gram '*pick up*' and remove the smaller ones '*pick*' and ''*up*'. Rules (3,4) filter out the larger incorrect n-grams. The intuition behind it is we do not want the robot to use more words than necessary to describe a concept, such as '*the red*' to describe the *red* colour. Rule (5) states that if the n-grams are connected to different concepts, keep all of them. This rule can be used to learn phrasal verbs where their meaning is different to their individual components. For example, the phrasal verb '*break down*' is different to both '*break*' and '*down*'. These rules will filter some of the incorrect associations. Also, they will not stop different synonyms from connecting to the same graphlet. For example, '*cyan*' and '*sky blue*' could share the same graphlet, because '*cyan*' is not a constituent of '*sky blue*. The robot ends up with a number of possible associations that require validation; details are in the next section.

Validation of Associations

Once strong associations have been generated (between n-grams and graphlets), we attempt to validate them by using the input video clip. For example, the 1-gram '*apple*' might have a high likely association with two different graphlets, one representing the shape graphlet *apple*, and the other representing something incorrect, e.g. the shape graphlet *mug*. This can occur due to noise or insufficient data, e.g. whenever the robot hears the word '*apple*' it finds a mug and an apple in the corresponding video clip.

The validation occurs by examining how these associations compare with the input videos. 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 creating multiple graph structures by connecting the graphlets together in all possible 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 (Howorka 1977), then we have validated the associations for these n-grams. For example, consider the sentence given in Fig. 4: "*pick up the apple*", and suppose that our robot is not sure of 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 'apple' 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 hypothesis graphs are generated that reflect all possible combinations. This is done by connecting the *connection nodes* (denoted with c) in both the action graphlet and the object graphlets together, shown in Fig. 6 (1, 2). We then check which (if any) of the generated hypothesis graphs match the input video. Since the hypothesis graph shown in Fig. 6-(1) 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 'apple' with their visual graphlets.

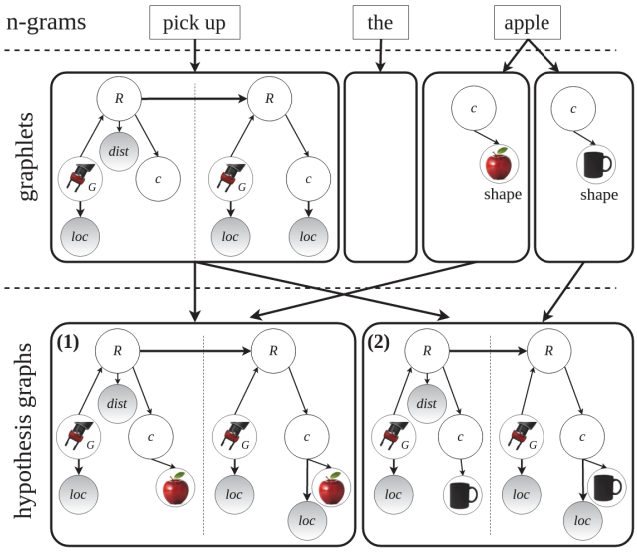


Figure 6: Generating *hypothesis graphs*.

Learning Grammar Rules

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

To acquire such grammatical knowledge, we use the correctly matched hypothesis graphs (described in the previous section) and generate their *syntactic trees*. These trees represent how n-grams should be labelled and connected (ordered) in the input sentence, based upon how their corresponding graphlets are connected in the matched hypothesis graph. The n-grams in the syntactic trees are labelled with a graphlet category (*colour, shape, location, direction, distance, action*), these labels are then grouped into a single semantic category (*S, object, spatial-relation, manipulated object (m-obj), function word (fw)*) in a similar way as presented in (Dukes 2013). The graphlet and semantic categories (e.g. *action*, *fw*) 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. As an example, for the command shown in Fig. 4: "*pick up the apple*", the correct hypothesis graph (graph (1) in Fig. 6) is used to generate its equivalent syntactic tree. The hypothesis graph encodes the knowledge of which objects are manipulated by the action in the input sentence. This information is mapped into a syntactic tree, as shown in Fig. 7. The n-grams in the input sentence are labelled to their corresponding graphlet types, e.g. 'pick up' is associated with an *action graphlet*, therefore is labelled *action*, 'apple' with a *shape graphlet* therefore is labelled *shape*, while 'the' has no associated graphlet therefore is labelled *fw*. After that, the object 'apple' is labelled as the manipulated object (*m-obj*) because the action 'pick up' was applied on the 'apple'.

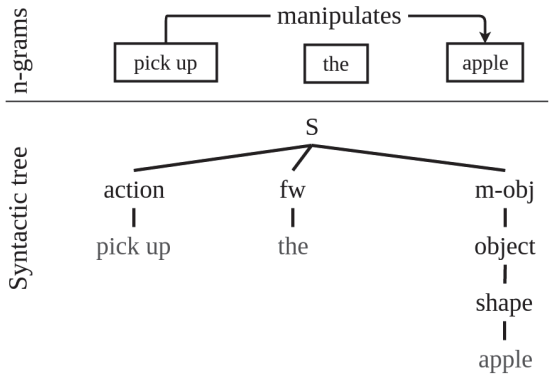


Figure 7: Example of a syntactic tree generated from the correctly matched hypothesis graph shown in Fig. 6 (1).

To learn the grammar rules from syntactic trees, we start 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 we observe. 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 learnt from only this example are shown in Table. 1, however, rules from all input examples are accumulated into one set.

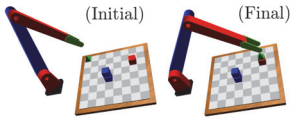
Learning Grammar Rules	
Grammar Rules	*Probability*
action → *pick up*	1.0
fw → *the*	1.0
shape → *apple*	1.0
S → *action, fw, m-obj*	1.0
m-obj → *object*	1.0
object → *shape*	1.0

Table 1: Learning grammar rules from syntactic tree in Fig. 7

Experimental Validation

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

For the **synthetic world**, we extended the *Train Robots* dataset which is designed to develop systems capable of understanding natural language commands (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 translated all the commands from English to a different language (Arabic), particular care was taken on not to alter any command or change any mistakes in any of them (the extended version is published at Alomari *et al.* (2016) http://doi.org/10.5518/32). We kept 200 videos and 1343 commands as our testing dataset. An example of the dataset is shown in Fig. 8.



EN: put the green pyramid on the red block
AR: ضع الهرم الأخضر فوق المكعب الأحمر

Figure 8: An Example from the *Train Robots* dataset, the Arabic sentence is translated from the English one.

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. The Kinect is used to collect RGBD videos of Baxter performing various manipulation tasks with real objects from the robot's point of view. We collected a dataset (http://doi.org/10.5518/110) consisting of 160 videos in which volunteers controlled the robot's arms, and manipulated real objects. 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 objects were manipulated in the videos such as basic block shapes, fruit, cutlery, and office supplies. The aim is to 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.

Evaluation

We evaluated the performance of our technique 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 using the learnt grammar.

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 main task is to learn the associations between words and feature space, e.g. the word 'red' associates to a graphlet containing the colour red. We define a correct matching by manually inspecting the matched n-gram-graphlet pairs and checking if the Gaussian component in the feature space falls within a single standard deviation of its typical values. E.g. red $\approx$ HSV$(0, 1, 1)$. Keeping in mind that our system starts with no pre-given knowledge, we evaluate the success of our technique based on the number of correctly matched n-grams and feature spaces. All the results presented in this section are computed using n-grams of $n \leq 3$.

Our system is able to correctly ground $47/53$ (88.6%) n-grams to their visual graphlets in the real-world test dataset, and $72/81$ (88.9%) in English and $90/101$ (89.1%) in Arabic in the synthetic test dataset, which clearly shows that our technique is able to ground words to visual features given raw linguistic and visual inputs. 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:

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

The system couldn't learn the visual representation of all n-grams in the datasets due to noise or lack of training data. For example, the n-gram *cyan* is 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.

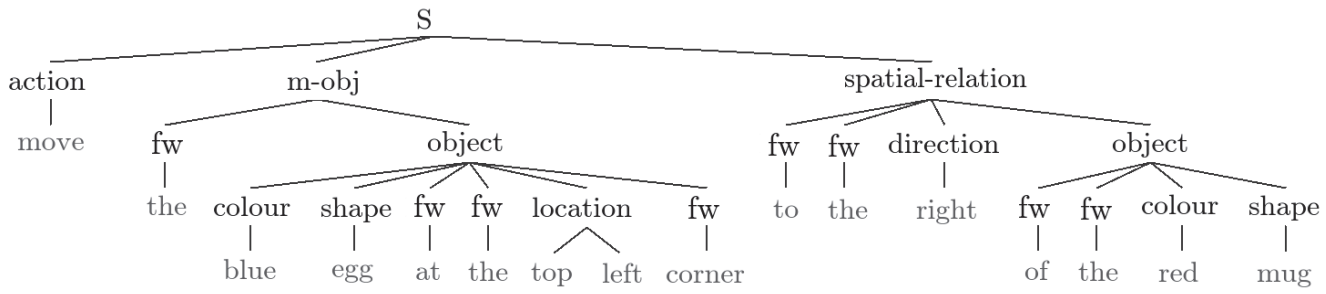


Figure 9: The syntactic tree generated for the new command "*move the blue egg at the top left corner to the right of the red mug*" using the knowledge gained from the training videos.

Parsing Novel Commands We also evaluate the system's ability to generalise its acquired knowledge to previously unseen objects and to parse new sentences/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 test videos, which we used to test 40 previously unseen (new) 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. 9. 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 133 grammar rules in the English, and 189 in the Arabic language were acquired, which we used to test on 1343 previously unseen commands (not included in training). In 929 (69.2%) of these commands the system was able to translate the command into a fully correct syntactic tree. Which is comparable with the state-of-the-art supervised system, standing at (90.7%) considering that we learn from unlabelled/raw data.

Conclusion and Future Work

We have demonstrated for the first time in a developmentally plausible setting, that a system can concurrently and incrementally learn two kinds of knowledge by processing language and vision real-world data: (i) the words' visual representations (actions, spatial-relations, and objects properties); and (ii) the probabilistic grammar rules. The spatio-temporal graphs (STDAG) 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. Also, the use of Gaussian components to represent visual features allows for efficient and incremental learning using the incremental Gaussian Mixture Model approach to update the different graphlets, which allowed us to represent all the input videos without the need to store the data points. We plan to extend our system to be able to learn from unsegmented videos and documents and to generate natural language descriptions from video clips using the gained/bootstrapped knowledge.

Learnt Grammar Rules	
Non-Terminals	
S → action, m-obj, spatial-relation	0.857
m-obj → fw, object	0.588
spatial-relation → 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
Terminal Leaves	
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. 9.

Acknowledgements

We thank colleagues in the School of Computing Robotics lab, in particular Majd Hawasly, and 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*.

Beetz, M.; Klank, U.; Kresse, I.; Maldonado, A.; Mosen-

lechner, L.; Pangercic, D.; Ruhr, T.; and Tenorth, M. 2011. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS*.

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

Dubba, K. S.; De Oliveira, M. R.; Lim, G. H.; Kasaei, H.; Lopes, L. S.; Tomé, A.; and Cohn, A. G. 2014. Grounding Language in Perception for Scene Conceptualization in Autonomous Robots. In *Qualitative Representations for Robots: Papers from the AAAI Spring Symposium.*

Duckworth, P.; Alomari, M.; Gatsoulis, Y.; Hogg, D. C.; and Cohn, A. G. 2016. Unsupervised activity recognition using latent semantic analysis on a mobile robot. In *22nd European Conf. on. Artificial Inteligence (ECAI)*.

Duckworth, P.; Alomari, M.; Charles, J.; Hogg, D. C.; and Cohn, A. G. 2017. Latent dirichlet allocation for unsupervised activity analysis on an autonomous mobile robot. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*.

Dukes, K. 2013. 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*.

Gatsoulis, Y.; Alomari, M.; Burbridge, C.; Dondrup, C.; Duckworth, P.; Lightbody, P.; Hanheide, M.; Hawes, N.; and Cohn, A. G. 2016. QSRlib: a software library for online acquisition of Qualitative Spatial Relations from Video. In *Workshop on Qualitative Reasoning (QR16), at IJCAI-16*.

Howorka, E. 1977. A characterization of distance-hereditary graphs. *The quarterly journal of mathematics* 28(4).

Huang, A. S.; Tellex, S.; Bachrach, A.; Kollar, T.; Roy, D.; and Roy, N. 2010. Natural language command of an autonomous micro-air vehicle. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2663–2669. IEEE.

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

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.

Lauria, S.; Bugmann, G.; Kyriacou, T.; and Klein, E. 2002. Mobile robot programming using natural language. *Robotics and Autonomous Systems* 38(3):171–181.

Matuszek, C.; Herbst, E.; Zettlemoyer, L.; and Fox, D. 2013. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*. Springer.

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

Muja, M., and Ciocarlie, M. 2013. tabletop object detector - ROS Wiki. http://www.ros.org/wiki/tabletop object detector.

Needham, C. J.; Santos, P. E.; Magee, D. R.; Devin, V.; Hogg, D. C.; and Cohn, A. G. 2005. Protocols from perceptual observations. *Artificial Intelligence* 167(1):103–136.

Parde, N.; Hair, A.; Papakostas, M.; Tsiakas, K.; Dagioglou, M.; Karkaletsis, V.; and Nielsen, R. D. 2015. Grounding the Meaning of Words through Vision and Interactive Gameplay. *Proceedings IJCAI 2015*.

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

Rusu, R. B. 2009. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. Ph.D. Dissertation, Computer Science department, Technische Universitaet Muenchen, Germany.

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

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*, volume 89.

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

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

Spranger, M., and Steels, L. 2015. Co-acquisition of syntax and semantics - an investigation in spatial language. In Yang, Q., and Wooldridge, M., eds., *IJCAI'15*. Palo Alto, US: AAAI Press. 1909–1905.

Spranger, M. 2015. 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*. 196–201.

Sridhar, M.; Cohn, A. G.; and Hogg, D. C. 2010. Unsupervised Learning of Event Classes from Video. In *AAAI*.

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

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.

Tellex, S.; Kollar, T.; Dickerson, S.; Walter, M. R.; Banerjee, A. G.; Teller, S.; and Roy, N. 2011. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine* 32(4):64–76.

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