From visuo-motor to language

Deepali Semwal, Sunakshi Gupta, Amitabha Mukerjee
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Abstract
We propose a learning agent that first learns concepts in an integrated, cross-modal manner, and then uses these as the semantics model to map language. We consider the action of throwing, considering the whole trajectory as a single image. A large set of such images, and the throwing parameters are mapped jointly onto a low-dimensional non-linear manifold. Such models improve with practice, and can be used as the starting point for real-life tasks such as aiming (e.g. dart throwing).

How can such models can be used in learning language? We consider a set of videos involving throwing and rolling actions. These actions are analyzed into a set of contrastive semantic classes based on the agent, action, and the thrown object (trajector). We obtain a crowdsourced commentaries for these videos (unannotated text) from a number of adults. A learner system attempts to learn labels using the contrastive probabilities for a given semantic class. Only a handful of high-confidence words are found, but the agent starts off with this partial knowledge. These are used to learn a potential set of syntactic patterns, for example for the trajector, and then for the agent-trajector-action sentences. We demonstrate how this may work for two completely different languages - English and Hindi, and also show how rudiments of agreement, synonymy and polysemy are detected.

Introduction
We claim two main contributions in this work: a. An unified process for mapping actions that involve visual changes (e.g. throwing, rolling, pulling, etc), b. Using this acquired motor model to map a given situation to language.

Mapping concepts multi-modally Consider the following situations:
(a) The agent recognizes [Sam throwing a ball to Sita]
(b) The agent understands the sentence “Sam threw the ball to Sita”

In classical AI, (a) would be modeled by a function whose input is visual and which classifies it as an action of class throw(); it would be trained on a large set of tagged videos. For the language input (b), one would parse the sentence into constituent phrases, and a semantic lexicon might map these to formal structures such as throw(agent:Sam, object: Ball, path: $p$)/goal($p$, Sita). It would be trained on a huge set of POS-, parse-, and semantically- annotated sentencees (Kwiatkowski et al. 2011), though the last may be avoided in some situations (Liang, Jordan, and Klein 2013). For (c), the robotic agent would command a sequence of motor torques at its joints. To learn to project the ball so that it goes to a target, it would train by throwing the ball repeatedly until it is able to achieve a desired path.

This process makes it hard to correlate knowledge between these modalities - e.g. looking at another person throwing short, it is hard to tell her to throw it higher, say. It is also inefficient in terms of requiring thrice the circuitry compared to a unified system. Indeed, primate brains seem to be operating in a more integrated manner. Motor behaviour invokes visual simulation to enable rapid detection of anomalies, while visual recognition invokes motor responses to check if they match. Linguistic meaning activates this wide range of modalities (Binder and Desai 2011). Such a cross-modal model also permits affordance and intentionality judgments, which our system can also achieve.

The first part of this work (section 2) is inspired by the observation that each time the robot throws the ball for motor learning), it also generates a visual trajectory from which it can learn a visual model of the action. Further, the motion parameters are correlated with the visual feedback - both lie on matched low-dimensional curved manifolds which can be aligned.

We develop an unified approach for modeling actions, based on visual inputs (say trajectories or paths) resulting from given motor commands. The model constitutes a low-dimensional manifold that discovers the correlations between visual and motor inputs. The model can be applied to either visual recognition or to motor tasks.
Bootstrapping a lexicon and syntax  For the language task (section 3), we consider the system which already has a rudimentary model for [throw], being exposed to a set of narratives while observing different instances of throwing. The narratives are crowdsourced from speakers for a set of synthetic videos which have different actions (throw or roll), agents (“dome” or “daisy” - male/female), thrown object or trajector (ball or square), colour of trajector (red or blue) and path (on target, or short, or long). We work with transcribed text, and not with direct speech, so we are assuming that our agent is able to isolate words in the input.

An important aspect of working with crowdsourced data is that for the very beginning learner, we need a more coherent input where similar phrases are used for similar situations. This is difficult, given the diversity of our crowdsourced input, so we first identify a small coherent subset (called the family lect) on which initial learning is done. This is then extended to the remaining narratives (the multi-lect).

The system works on subsets of the narratives for each semantically distinct category. The joint word-semantics probabilities are computed. To learn the label for [ball], we contrast the frequency of a word being uttered when a [ball] is thrown or rolled, vs a [square]. Candidate labels are ranked based on the ratio of these joint probabilities - $p(\text{word,concept}) / p(\text{word,non-concept})$. The high-confidence matches - those significantly higher than the next match (about 20%) are taken as the initial bootstrapping map (fig. 4). This partial lexicon is then used to learn a partial syntax, which is ploughed back to learn more lexemes (and also synonyms and alternations). When this interleaving stabilizes, we broaden the semantic context to learn other structures. Finally, we find that we are able to discover a good chunk of transitive verb syntax. The system is demonstrated on two very different languages, English and Hindi. For Hindi we find that are also able to discover some agreement relations such as gender.

This partial-analysis based approach is substantially different from other attempts at grounded modeling in NLP, which have focused on demonstrating the acquisition of syntax /morphosyntax (Madden, Hoen, and Dominey 2010), (Kwiatkowski et al. 2011), (Nayak and Mukerjee 2012). One may call this approach dynamic NLP, since it keeps learning from every sentence, and does not generate a static model. Also, after an initial bootstrapping phase driven by this multi-modal corpus, learning can continue to be informed by text alone, a process well-known from the rapid vocabulary growth after the first phase of language acquisition in children (Bloom 2000).

Visuo-motor Pattern Discovery
Learning a few visuo-motor tasks are among our agent’s very first achievements. Let us consider the act of throwing a ball. Our learner knows the motor parameters of the throw as it is being thrown - here we focus not on the sequence of motor torques, but just the angle and velocity at the point of release.

Each trajectory gives us an image (samples - fig. 1). We are given a large set of images (say, N=1080), each with 100×100 pixels. Each image can be thought of as a point in a $10^4$-dimensional space. The set of possible images is enormous, but we note that if we assign pixels randomly, the probability that the resulting image will be a [throw] trajectory is practically zero. Thus, the subspace of [throw] images is very small.

Next we would like to ask what types of changes can we make to an image while keeping it within this subspace? In fact, since each throw varies only on the parameters $(\theta, v)$, there are only two ways in which we can modify the images while remaining locally within the subspace of throw images. This is the dimensionality of the local tangent space at any point, and by stitching up these tangent spaces we can model the entire subspace as a non-linear manifold of the same intrinsic dimensionality. The struture of this image manifold exactly mimics the structure of the motor parameters (the motor manifold). They can be mapped to a single joint manifold, which can be discovered using standard non-linear dimensionality reduction algorithms such as ISOMAP. In fig. 2, we show the resulting manifold obtained using a hausdorff distance metric (Huttenlocher, Klanderman, and Rucklidge 1993): $h(A, B) = \max_{a \in A, \min_{b \in B} \|a - b\|}$.

![Figure 2: variations in the manifold according to a) angle of projection, and b) velocity. (low values in yellow)](image)

The same idea can be used to find correlations in any system involving a motion or a change of initial conditions, using this algorithm:

**Algorithm 1** Visuo-motor law discovery Algorithm
1: **Input**: Set of high dimensional images $\{I_1, I_2, I_3, ..., I_N\}$, and corresponding control parameters.
Table 1: Sum Absolute Error (Velocity) falls as N increases

<table>
<thead>
<tr>
<th>No. of images</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>600</th>
<th>800</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAE in velocity</td>
<td>3.69</td>
<td>3.57</td>
<td>3.85</td>
<td>2.71</td>
<td>2.38</td>
<td>1.93</td>
</tr>
</tbody>
</table>

2: **Step1:** Obtain a low dimensional embedding for the images using ISOMAP.
3: **Step2:** Train a regression model to acquire the mapping from the low dimensional (curved) coordinates to the control parameters.
4: **Step3:** For executing a new throw, use a (query) image with desired path. Find a linear interpolation $J$ for this query image: $J = \sum_{j=1}^{k} w_j I_j$
5: **Step4:** Calculate the embedding points for the query image using the weights learnt in Step3. $Q = \sum_{j=1}^{k} w_j q_j$
6: **Step5:** Use the mapping learnt in Step2 to obtain the corresponding parameters for the query image $J_i$.

end

This is a generic algorithm for discovering patterns in visuo-motor activity. Initially, its estimate of how to achieve a throw are very bad, but they improve with experience. We model this process in table 1 - as the number of inputs $N$ increases, the error in predicting a throw decreases - at N=100, the error is nearly twice that at N=1000.

Figure 3: *Throwing darts*. a) Two trajectories that hit the board near the middle. b) 2D manifold of projectile images, showing band for “good throws”, and interpolated curve for “best” throws.

The visuo-motor manifold model developed here is not task-specific, but can be applied for many tasks involving projectile motion, catching a ball, throwing at a basket, darts, tennis, etc. As an example we can consider darts - ignoring the lateral deviations, the successful trajectories are those that intersect the dartboard near its center (fig. 3a). This corresponds to a “good zone” in the latent space (central band in fig. 3b) Of these, one may wish to select those that are nearly orthogonal to the board at contact (short curved axis). Of course, actual task performance will improve with experience (more data points in the vicinity of the goal), resulting in better performance. This model only provides a starting point.

Figure 4: High confidence lexeme discovery: Contrastive scores and dominance ratio for top lexemes in. Highlighted squares show high-confidence lexemes (ratio more than twice)

**Bootstrapping Language**

In order to learn language, we create a set of 15 situations involving the actions [throw] and [roll] (one of the agents is shown in fig. 1. The videos of these actions are then put up for commentary. Largely students on the campus network contributed; we obtain 18 Hindi and 11 English transcribed narratives to work on.

The compiled commentaries vary widely in lexical and constructional choices. An example description for a video is “daisy throws a blue square”. Another subject describes the same video as “now daisy threw the blue box which fell right on the mark”. Both these narratives varies in terms of lexical units used as well the details incorporated in description.

As described earlier, we first select a coherent subset of narratives - those that have a more consistent vocabulary from - and identify these as the family lect.

Given the [throw] model, the system can identify the act of throwing, and also the agent who throws, the object thrown, and its path. Further, we assume similar capabilities (not implemented) for [roll], and also the ability to discriminate a square from a circle, red from blue, and the two agents - [Dome] with a moustache (fig. 1, male), and [Daisy] with a ponytail. Based on these distinctions, it tries to find words that differ in their usage between the two contrasting situations. [Note: we use square brackets to indicate a concept, the semantic pole of a linguistic symbol] This contrastive approach has been suggested as a possible strategy applied by child learners (Markman 1990). Given two contrasting concepts $c_1, c_2$, we compute the empirical joint probabilities of word (or later, n-gram) $\omega_i$ and concept...
c₁, c₂, and compute their contrast score:
\[ S_{\omega₁,c₁} = \frac{P(\omega₁,c₁)}{P(\omega₁,c₂)} \]

We also compare the corpus of narratives here with that from a larger unannotated corpus (Brown Corpus for English and the CIL/ITB Corpus for Hindi). We look for words that are more frequent in the domain input than in a general situation. This rules out many frequent words like a, and, the for English, and है, क्र (hai, ke) [is, of] (Hindi). The small set of high confidence words - whose contrastive probability is more than twice the next best match - are highlighted in (Fig. 4).

**Interleaving of Word / Syntax learning**

Once the system has a few grounded lexemes, we proceed to discovering syntactic constructions. At the start, we try to learn the structure of small contiguous elements. One assumption we use here is that concepts that are very tightly bound (e.g. object and its colour) are also likely to appear in close proximity in the text (Markman 1990). Another assumption (often called syntactic bootstrapping), is used for mapping new phrases and creating equivalence classes or contextual synonyms. This says that given a syntactic pattern, if phrase p₁ appears in place of a known phrase p₀, and if this substitution is otherwise improbable (e.g. the phrase is quite long), then p₀,p₁ are synonyms (in the same semantic class) or they are in the same syntactic lexical category.

We find that one type of trajector (e.g. “ball”) and its colour attribute (e.g. “red”, [red]) have been recognized. So the agent pays more attention to situations where these words appear. Computationally this is modelled by trying to find patterns among the strings starting and ending with (delimited by) one of these high-confidence labels (e.g. “red coloured ball”). This delimited corpus consists of strings related to a known trajector-attribute complex. Within these tight fragments, we show that standard grammar induction procedures are able to discover preliminary word-order patterns which can be used to induce broader regularities. We compare two available unsupervised grammar induction systems - Fast unsupervised incremental parsing (Seginer 2007) and ADIOS (Solan et al. 2005); results shown here adopt the latter because of a more explicit handling of discovered lexical classes.

The initial patterns learned for the trajector in this manner are shown in Table 2. These are generalized using the phrase substitution process to yield the new lexeme बॉल (ball), [ball], in Hindi and “blue” in English (Table 3). This is done based on the family-lect (Figure 4), and the filtered sub-corpus is used to learn patterns and equivalence classes.

The system now knows patterns for [red square], say, and it now pays attention to situations where the pattern is almost present, except for a single substitution. It can look into a the other semantic classes as well, (e.g. [blue square], [red ball]). In most of these instances (Fig. 4) we already have partial evidence for these units from their contrastive scores. Now if we discover new substitution phrases p₁ in the position of p₀, referring to a concept in the same semantic class (e.g. [ball] for [square]), and if p₁ is already partially acceptable for [ball] based on contrastive probability, then p₁ becomes an acceptable label for this semantic concept. This process iterates - new lexemes are used to induce new patterns, and then further new lexemes, until the patterns stabilize. This is then extended to the entire corpus beyond the small family lect; results are shown in Table 4.

The table captures a reasonable diversity of Noun Phrase patterns describing coloured objects. Note that words like “red” and “ball” have also become conventionalized in Hindi. We also observe that the token niley appears in the 4-word pattern niley rang kaa chaukor and is highly confident even from a single occurrence; this reflects the fast mapping process observed in child language acquisition after the initial grounding phase (Bloom 2000). As the iteration progresses, these patterns are used for further enhancing the learners inventory of partial grammar.

**Verb phrases and sentence syntax**

Having learned the syntax for a trajector, this part of the input is now known with some confidence, and the learner can venture out to relate the agent to the action and path. In this study, we failed to find any high-confidence lexemes related to path, hence we were not able to bootstrap that aspect. In the following we restrict ourselves to patterns for the semantic classes [agent], [action], [trajector].

At this stage, the agent notes that many of the words seem rather similar (e.g. “sarkAyA”, “sarkAyI” (H); or “throwing”, “thrown” (E)). A text-based morphological similarity analysis is reveals several clusters with alterations at the end of words (Fig. 5). To quantify this aspect, we consider normalised Levenshtein distance and

<table>
<thead>
<tr>
<th>Hindi : लाल बॉल (haal chaulko) [red square] नीले रंग का चौकोर (haal rang ka chaukor) [red coloured-GEN square]</th>
<th>English : red → ball square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi : लाल बॉल (haal ball) [red ball] नीले रंग का चौकोर (blue coloured-GEN square)</td>
<td>English : blue → ball square</td>
</tr>
</tbody>
</table>

Table 2: Initial constructions learned from the trajector delimited strings

Table 3: Syntax for trajector - iteration 1, Family-lect
perform a morphological similarity analysis. Since our input is text, we limit ourselves to analysis based on the alphabetic patterns as opposed to phonemic maps. Similar words are clustered using a normalized similarity index. Thus we have twelve type instances of (।-aa) - (।-ii), and seven types for (।-aa) - (।-e). We find that these variants - e.g. “sarkaayaa”, “sarkaayii” - appear in the same syntactic and semantic context. These clusters are now used to further strengthen the lexeme and action association.

Again, we use an iterative process, starting with grounded unigrams, moving a level up to learn simple word-order patterns, learning alternations and lexical classes through phrase substitution, and so on to acquire a richer lexicon and syntax. The learner has the concept of agent and has associated the words “daisy” and “dome”. One action word is known for both Hindi and English (Fig. 5), and these are used to obtain a filtered corpus. To simplify the task of discovering verb phrase syntax, the known trajector syntax (table 4) is considered as a unit (denoted as [TRJ], and the concept of agent ([daisy] or [dome]) is denoted [AGT].

Results of initial patterns, obtained based on the two known action lexemes, are shown in Table 5. Note that pattern matching with Adios now discovers the class a, the as equivalence class E23 (replaced hereon).

Next, we interleave this syntactic discovery with lexical discovery, permitting also bigram substitutions. Thus “threw” and “rolled” are found to be substitutable by has thrown, is throwing; and has slid respectively. This gives us the more general results of Table 6. Again the system iterates over the corpus till the discovery of new patterns converges (Table 7) results. An interesting observation is that the Hindi data finds a phrase in the TRJ position - नीला चौकोर (nilaa chaukor) [blue square]. This had not been learned in the trajector iteration, since nilaa was less frequent.

Thus we see that with this approach we are able to acquire several significant patterns. These patterns apply to only a single action input, and for a very limited set of other participants. But it would be reasonable to say that the agent may observer similar structures elsewhere - e.g. in a context involving hitting, say, if we have the sentence “Daisy hit Dome” then the agent may use the syntax of [AGT] [verb] [TRJ] to extend to this context and guess that “hit” may be a verb and “Dome” the object of this action (which it knows from the semantics). Thus, once a few patterns are known, it becomes easier to learn more and more patterns, which is the fast mapping stage we have commented upon ear-
A second area would be to extend this work in terms of greater depth. Consider sentences such as (d) “Sam threw the searchlight beam at the roof.”, or (e) “Sam threw her a searching look.” These are relatable to the meaning of [throw] used here, at the model level. But these are typically never encountered in a grounded manner, but very often in language-only input. Linking this language input to the existing grounded models would open up new vistas for concept and language discovery.

**References**


