



HAL
open science

Experiments using semantics for learning language comprehension and production

Dana Angluin, Leonor Becerra-Bonache

► **To cite this version:**

Dana Angluin, Leonor Becerra-Bonache. Experiments using semantics for learning language comprehension and production. Bio-inspired Models for Natural and Formal Languages, Cambridge Scholars Publishing, pp.3-32, 2011. hal-00620729

HAL Id: hal-00620729

<https://hal.science/hal-00620729>

Submitted on 8 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Experiments Using Semantics for Learning Language Comprehension and Production

Dana Angluin*

Leonor Becerra-Bonache†

Abstract

Several questions in natural language learning may be addressed by studying formal language learning models. In this work we hope to contribute to a deeper understanding of the role of semantics in language acquisition. We propose a simple formal model of meaning and denotation using finite state transducers, and an algorithm that learns a meaning function from examples consisting of a situation and an utterance denoting something in the situation. We describe the results of testing this algorithm in a domain of geometric shapes and their properties and relations in several natural languages: Arabic, English, Greek, Hebrew, Hindi, Mandarin, Russian, Spanish, and Turkish. In addition, we explore how a learner who has learned to comprehend utterances might go about learning to produce them, and present experimental results for this task. One concrete goal of our formal model is to be able to give an account of interactions in which an adult provides a meaning-preserving and grammatically correct expansion of a child's incomplete utterance.

1 Introduction

Most research in the field of Grammatical Inference has focused on learning syntax, and tends to omit any semantic information. However, linguistic and cognitive studies suggest that semantic and pragmatic information are available to children learning language (e.g., [9, 14, 21]). It appears that semantics and context play an especially important role in the 2-word stage of child linguistic development, in which children go from the production of one word to the combination of two elements, and syntax first becomes relevant. In this stage, context is important to understand the meaning of 2-word sentences and, thanks to the shared context, child and adult can communicate with each other although their grammars are different.

Taking into account that formal language learning is hard (as results obtained in Grammatical Inference show [10]), and it seems more natural to take into account semantics in language learning, the following question arises: can semantic information simplify the problem of learning grammar? Our conjecture is that semantics can make learning easier. In the fields of Linguistics and Cognitive Science there is a sizable body of research on learning semantics, but there has been little work on the use of semantic information to guide the acquisition of syntax (e.g., [13, 20]).

Inspired by the 2-word stage, we propose a simple computational model that takes into account *semantics* and *context* for language learning. In contrast to other approaches [12, 15, 24], our model does not rely on a complex syntactic mechanism. In our model, the shared semantic situation

*Department of Computer Science, Yale University, USA. Research supported by National Science Foundation Grant CCF-0916389.

†Department of Computer Science, Yale University, USA, and Research Group on Mathematical Linguistics, Rovira i Virgili University, Spain. Supported by a Marie Curie International Fellowship within the 6th European Community Framework Programme.

allows communication between child and adult, despite the fact that their grammars may be very different. Also, the inputs to our learning algorithm are utterances and the situations in which these utterances are produced, rather than utterances and their meanings, as assumed in other studies. In this respect our model captures one aspect of the 2-word stage: in addition to hearing utterances, children have access to the context in which those utterances are generated.

Our model is also designed to address the issue of the kinds of input available to the learner. Positive data is clearly an essential part of the process of language acquisition and plays the main role in that process. However, we also want to model another kind of information that is specifically available to the child during the 2-word stage, but which has not generally been taken into account in formal models of language acquisition. It is called *expansion*. Consider the following example (extracted from [6]):

CHILD: Baby highchair
ADULT: Baby is in the highchair

As we can see, the adult’s answer is an expansion of an incomplete sentence uttered by the child. Therefore, a correction is given to the child by means of a *meaning-preserving expansion* (in which the child’s utterance and the adult’s correction have the same meaning, but different forms.) The context also plays an important role: adult and child share the context, and the adult uses the semantic situation in order to correct the child. Chouinard and Clark studied such corrections in [9]. They analyzed longitudinal data from five children between 2;0 and 4;0, and showed that: (i) adults reformulate erroneous child utterances often enough to help learning, and (ii) children can detect and make use of differences between their own utterance and the adult reformulation. Thus, in natural situations, corrections have a *semantic* component that has not been taken into account in previous Grammatical Inference studies. One of our long-term goals is to find a formal model that gives an account of this kind of correction in which we can address the following questions: What are the effects of corrections on learning syntax? Can corrections facilitate the language learning process?

Moreover, corrections seem to have a close relation to positive data because of semantics in a shared context. Consider the two situations depicted in Figure 1. In case A, the child receives an utterance from the adult, uses the context to determine the meaning, and then re-expresses it in her own grammar.¹ In case B, the child produces a sentence in her own grammar, the adult determines the meaning using the context and then supplies an expansion to the child. From the response, the child can see whether the adult misunderstands her message, and how it can be expressed in the adult’s grammar.

As we can see, positive data and corrections have similar effects on the child. With either positive data or corrections: the child can see how to express a situation using a correct sentence (with respect to the adult’s grammar); the production of the child is like an echo of the corresponding adult sentence; although the grammars of the adult and the child are different, they can communicate with each other thanks to the semantics. Hence, a model that takes into account semantics may give us a better understanding of the relation between positive data and corrections.

Here we present a computational model that takes into account semantics for language learning. We focus initially on a simple formal framework, which we hope will extend to one with more cognitive plausibility. We note that our model is inspired by natural language acquisition studies, but is not directly intended as a realistic model of how children or adults learn language. In particular, we neglect many important aspects of language learning, including realistic sensory input and embodiment, biologically plausible computation, segmentation, phonology, morphology,

¹Sometimes children explicitly verbalize this as a kind of reduced “echo” of the adult’s sentence [14].

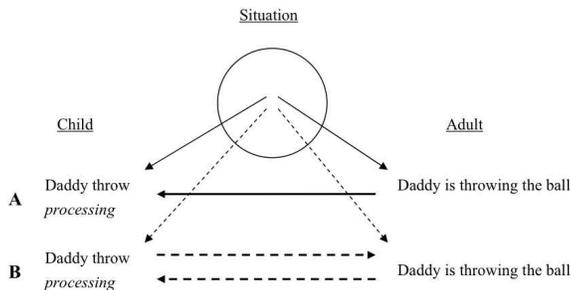


Figure 1: Adult and child share the same situation. In case A, child receives positive data. In B, a correction is returned to the child.

complex syntax, and pragmatics. The fragments of natural language we consider are very simple, and were chosen to be amenable to representation by finite state transducers. Our goal is to investigate aspects of the roles of semantics and corrections in the process of language learning. The current paper should be seen as an initial step towards this goal.

Our model accommodates two different tasks: comprehension and production. The scenario we consider is cross-situational and supervised, i.e., the teacher provides to the learner several example pairs consisting of a situation and a utterance that denotes something in the situation. For comprehension, the goal of the learner is to learn the meaning function, which allows the learner to comprehend novel utterances. Then, assuming the learner has learned the meaning function, we explore one model of how the learner could move from telegraphic speech to adult speech.

In Section 2, we describe the meaning and denotation functions used by the teacher to provide examples to the learner. In Section 3, we describe an algorithm to learn a meaning function. In Section 4, we present and analyze the results of testing our algorithm to learn a meaning function with samples of several natural languages in a restricted domain. In Section 5, we present a model of how the learner could move from telegraphic to adult speech, based on the idea of translating a meaning into an utterance. In Section 6, we present experimental results for this approach to learning production. Final remarks and future work are in Section 7. More details are available in [2, 4]

2 Meaning and Denotation Functions

To specify a meaning function, we use a finite state transducer M that maps sequences of words to sequences of predicate symbols, and a path-mapping function π that maps sequences of predicate symbols to sequences of logical atoms. We define situations and the concept of a match of a sequence of atoms in a situation, and use these to define the denotation of a sequence of words in a situation.

2.1 Meaning transducers

We consider three disjoint finite alphabets of symbols: W , the set of **words**, P , the set of **unary predicate symbols**, and R , the set of **primary binary predicate symbols**.

For each symbol $r \in R$, there is also a new binary predicate symbol r^t , which is used to denote r with its arguments reversed; the set of all such r^t is denoted R^t . The symbols in P and R are **primary predicates**, and the symbols in R^t are **derived predicates**. The function *primary* maps each primary predicate symbol to itself, and each predicate symbol r^t to r .

An **utterance** is a finite sequence of words, that is, an element of W^* . Define the function c to map a finite sequence of elements to the set of distinct elements occurring in the sequence. Thus, $c(u)$ is the set of words occurring in the utterance u .

We define a **meaning transducer** M with input symbols W and output symbols $Y = P \cup R \cup R^t$. M has a finite set Q of states, an initial state $q_0 \in Q$, a finite set $F \subseteq Q$ of final states, a deterministic transition function δ mapping $Q \times W$ to Q , and an output function γ mapping $Q \times W$ to $Y \cup \{\varepsilon\}$, where ε denotes the empty sequence. Note that a meaning transducer is deterministic.

The transition function δ is extended to define $\delta(q, u)$ to be the state reached from q following the transitions specified by the utterance u . The **language** of M , denoted $L(M)$ is the set of all utterances $u \in W^*$ such that $\delta(q_0, u) \in F$. For each utterance u , we define the **output** of M , denoted $M(u)$, to be the finite sequence of non-empty outputs produced by starting at state q_0 and following the transitions specified by u . A state $q \in Q$ is **live** if there exists an utterance u such that $\delta(q, u) \in F$, and **dead** otherwise.

As an illustration, we describe an extended example of utterances in English involving geometric shapes and their properties and relative locations. W contains the words *the, triangle, square, circle, red, blue, green, above, below, to, left, right, and of*. P contains the symbols *tr, sq, ci, bi, re, bl, gr* referring to the properties of being a triangle, a square, a circle, big, red, blue, and green, respectively, and R contains the symbols *ab* and *le*, referring to the relations of being above and to the left of, respectively. (Note that there is no word *big* – a property or relation may not have a corresponding word.) We define the meaning transducer M_1 as follows. M_1 has states q_i for $0 \leq i \leq 7$; q_0 is the initial state and there is one final state, q_2 . Figure 2 depicts the transducer M_1 . The dead state, q_7 , and transitions to it are omitted.

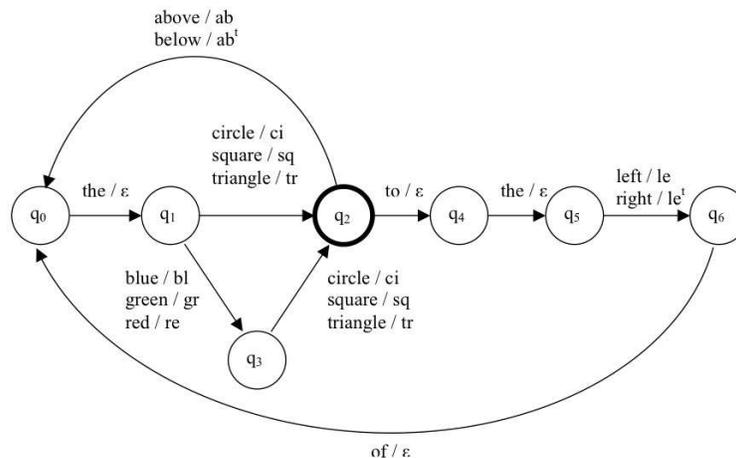


Figure 2: Meaning transducer M_1

$L(M_1)$ contains utterances such as *the triangle*, *the blue triangle to the left of the red circle*, and *the circle to the left of the green triangle above the blue square*.² The output of M_1 for the utterance *the triangle* is just the sequence (tr) , because the empty output for *the* is omitted. The output of M_1 for the utterance *the blue triangle above the square* is the sequence (bl, tr, ab, sq) .

²Having more than two objects in such an utterance is somewhat artificial, but it allows us to define an infinite language.

2.2 Path-mapping

Given a finite sequence of predicate symbols, we define a specific function, path-mapping, to convert it into a finite sequence of atoms in the predicate logic. Let x_1, x_2, \dots be distinct variables and t_1, t_2, \dots be distinct constants. Different constants will be used to denote different objects in a situation. An **atom** is one of $p(v)$, where $p \in P$, or $r(v, w)$ or $r^t(v, w)$, where $r \in R$ and v and w are constants or variables. An atom is **primary** if its predicate symbol is primary, that is, not from R^t . An atom is **ground** if it does not contain any variables.

The **path-mapping function**, denoted π , takes a finite sequence of predicate symbols and supplies each predicate with the correct number of argument variables, as follows. Taking the predicate symbols in order, x_1 is the argument of each predicate in the initial sequence of unary predicates, then x_1 and x_2 (in order) are the arguments of the first binary predicate, then x_2 is the argument of each of the subsequent sequence of unary predicates, then x_2 and x_3 (in order) are the arguments of the second binary predicate, and so on, introducing successive variables for successive binary predicates. Applying π to the sequence of predicates

$$(bl, tr, ab, sq, le^t, re, ci),$$

we get the following sequence of atoms

$$(bl(x_1), tr(x_1), ab(x_1, x_2), sq(x_2), le^t(x_2, x_3), re(x_3), ci(x_3)).$$

The **meaning** assigned by a meaning transducer M to an utterance u is $\pi(M(u))$. As an example, the meaning assigned by M_1 to the utterance *the blue square to the right of the green circle* is

$$(bl(x_1), sq(x_1), le^t(x_1, x_2), gr(x_2), ci(x_2)).$$

Because the path-mapping function π is fixed, knowing $M(u)$ is sufficient to compute the meaning $\pi(M(u))$. The definition of π reflects a strong restriction on the way properties and relations can be expressed by a meaning transducer. (In particular, this condition is not satisfied by our Hungarian sample, in which the preposition for the meaning *ab* appears after both of its arguments.)

2.3 Situations and denotation functions

A **situation** is a finite set of primary ground atoms. Situations represent the objects, properties and binary relations that are noticed in some environment by the teacher and learner. Only primary predicates (from $P \cup R$) occur in situations, although meanings may use both primary predicates and derived predicates (from R^t).

For example, noticing a big blue triangle above a big green square gives the following situation.

$$S_1 = \{bl(t_1), bi(t_1), tr(t_1), ab(t_1, t_2), gr(t_2), bi(t_2), sq(t_2)\}.$$

The **things** in a situation S , denoted $things(S)$, is the set of all t_i that occur in atoms in S . The assignment of constants t_i to things in the situation is arbitrary, as long as different things are represented by different constants. The **predicates** in a situation S , denoted $predicates(S)$, is the set of all predicate symbols that occur in atoms in S .

To determine the denotation of an utterance u in a situation S , we take the meaning $\pi(M(u))$ of u and attempt to match it in the situation. A ground atom A is **supported by** a situation S if A is primary and an element of S , or, if A is $r^t(t_i, t_j)$ for some $r \in R$ and $r(t_j, t_i)$ is an element of S . For example, $gr(t_2)$, $ab(t_1, t_2)$, and $ab^t(t_2, t_1)$ are supported by the situation S_1 defined above, but $gr(t_1)$ and $le(t_1, t_2)$ are not.

Let $V = \{x_1, \dots, x_k\}$ denote the variables that occur in $\pi(M(u))$ and T denote the things in the situation S . A **match** of $\pi(M(u))$ in S is a one-to-one function f from V to T such that substituting $f(x_i)$ for every occurrence of x_i in $\pi(M(u))$ produces a set of ground atoms that are all supported by the situation S . A match is **unique** if no other one-to-one function of V to T is also a match of $\pi(M(u))$ in S . Given a match f , the **first thing mentioned** is the constant $f(x_1)$ and the **last thing mentioned** is the constant $f(x_k)$.

As an example, the function $f(x_1) = t_1$ and $f(x_2) = t_2$ is a unique match of $\pi((bl, tr, ab, sq))$ in the situation S_1 . In this match, the first thing mentioned is t_1 and the last thing mentioned is t_2 . If we consider the utterance *the square below the blue triangle*, the output of M_1 is (sq, ab^t, bl, tr) . The function $g(x_1) = t_2, g(x_2) = t_1$ is a unique match of π applied to this output in S_1 , and in this case, the first thing mentioned is t_2 and the last thing mentioned is t_1 .

A **denotation function** is specified by a meaning transducer M and a choice of a parameter *which* from $\{first, last\}$. Given an utterance u and a situation S such that $u \in L(M)$ and there is a unique match f of $\pi(M(u))$ in S , then the denoted object is the first thing mentioned if *which* = *first* and the last thing mentioned if *which* = *last*. Otherwise, the denotation function is undefined for u and S .

With M_1 we specify a denotation function by choosing *which* = *first*. Then in the situation S_1 , the utterance *the blue triangle above the square* denotes t_1 and *the square below the blue triangle* denotes t_2 . The utterance *the green triangle* has no denotation in the situation S_1 .

3 The Algorithm to Learn a Meaning Function

We make several assumptions about the meaning transducer and the input data to simplify the task of the learner. Assumption 1 is that the output function $\gamma(q, w)$ does not depend on the state q ; thus we write $\gamma(w)$. This assumption holds of the transducer M_1 and the corresponding transducers for the other languages in our empirical study. It facilitates a strategy based on cross-situational conjunctive learning.

We also extend γ to map sequences of words to sequences of predicates by mapping each word in order and concatenating the results. Thus, for every sequence of words u , $\gamma(u) = M(u)$. We may think of M as consisting of an acceptor for $L(M)$ and a separate output function γ .

In learning to comprehend the denotations of utterances in $L(M)$, the learner depends on the teacher to produce an utterance that is grammatically correct and that denotes something in their shared situation. This frees the learner to concentrate on learning γ .

Once γ is learned, learning the denotation function is a matter of correctly setting the parameter *which*; for this part of the task, we assume that the learner receives a nonverbal indication of which thing is denoted, for example, the teacher points at it.

The input to the learning algorithm is a sequence of pairs (S_i, u_i) where S_i is a situation and u_i is an utterance denoting one of the things in the situation S_i . The goal of the learning algorithm is to learn a mapping γ' such that $\gamma'(u) = \gamma(u)$ for all $u \in L(M)$. We describe how the algorithm chooses γ_n , its hypothesis for γ' after the first n input pairs. Although for ease of understanding we describe the algorithm as storing all n input pairs seen so far, it is not difficult to update the data structures incrementally to avoid this.

1. Let G_n denote $\bigcap_{i=1}^n predicates(S_i)$, the set of primary predicates that have occurred in every situation so far. (These are the current background predicates. It is assumed that predicates that are denoted by words will be present in some situations and absent in others, so predicates are ignored if they have occurred in every situation so far.)

2. Let W_n denote the set of all words encountered so far. Form the partition \mathcal{K}_n of W_n according to the function $h(w) = \{i : w \text{ occurs in } u_i\}$. (The blocks of the partition are the current *word co-occurrence classes*. The words in a co-occurrence class occur in exactly the same set of utterances so far.)
3. For each word co-occurrence class $K \in \mathcal{K}_n$, let

$$U_n(K) = P \cap \bigcap_{i=1}^n \{\text{predicates}(S_i) : K \subseteq c(u_i)\}.$$

(This is the set of unary predicates that occur in every situation in which K occurred; they are the unary candidates for the meaning of K .)

If $(U_n(K) - G_n) \neq \emptyset$ then choose one word $w \in K$ and one predicate $p \in (U_n(K) - G_n)$, and define $\gamma_n(w) = p$ and $\gamma_n(w') = \varepsilon$ for each $w' \in K$ such that $w' \neq w$. (This assigns one non-background unary predicate to each word co-occurrence class with any unary candidates.)

4. For each $K \in \mathcal{K}_n$ such that $(U_n(K) - G_n) = \emptyset$ define

$$B_n(K) = \bigcap_{i=1}^n \{\text{possible}(S_i, u_i) : K \subseteq c(u_i)\}.$$

(This is all the binary predicates from $R \cup R^t$ that are possible meanings of K , given the unary predicates assigned by γ_n ; these are the binary candidates for the meaning of K . The possibility relation is described below.)

If $(B_n(K) - G_n) \neq \emptyset$ then choose one word $w \in K$ and one binary predicate $r \in (B_n(K) - G_n)$, and define $\gamma_n(w) = r$ and $\gamma_n(w') = \varepsilon$ for each $w' \in K$ such that $w' \neq w$. (This assigns one non-background binary predicate to each word co-occurrence class not already assigned a unary predicate that has any possible binary candidates.)

5. For each word w not yet assigned a value by γ_n , define $\gamma_n(w) = \varepsilon$, and output γ_n , the meaning function conjectured after n input pairs.

To complete this description, we need to specify how $\text{possible}(S_i, u_i)$ is defined, given γ_n as partially defined in step (3). Apply γ_n to u_i , treating undefined values as ε , to get a sequence (p_1, p_2, \dots, p_k) of unary predicates. With respect to this sequence of predicates, a sequence $(t_{i_1}, t_{i_2}, \dots, t_{i_r})$ of things is a **possible ordering of things** if there is a partition of $(1, 2, \dots, k)$ into r (possibly empty) consecutive intervals I_1, I_2, \dots, I_r such that for each $j = 1, 2, \dots, r$, the atom $p_j(t_{i_s})$ is supported by the situation S_i for all $i_s \in I_j$. That is, there must be a way to assign the unary predicates to the things (in order) so that the predicates hold of the things in the situation. A binary predicate $r \in R$ is in $\text{possible}(S_i, u_i)$ if there is an atom $r(t_r, t_s) \in S_i$ and a possible ordering of things in which t_r immediately precedes t_s . Analogously, a binary predicate $r^t \in R^t$ is in $\text{possible}(S_i, u_i)$ if there is an atom $r(t_r, t_s) \in S_i$ and a possible ordering of things in which t_s immediately precedes t_r . In this case, correctly learning the meaning function for unary predicates constrains the possible choices for binary predicates under the assumption that the word(s) denoting a binary predicate separate the word(s) denoting its arguments in an utterance.

An example of the results for a random sample of six pairs (S_i, u_i) , where u_i is an utterance in Spanish, is presented in Table 1. Situations are abbreviated, for example, *bgsלבgt* denotes a big green square to the left of a big green triangle. As we can see, there is just one background predicate: *bi*. The algorithm forms nine different word co-occurrence classes and for each one, intersects the set

of primary predicates occurring in their corresponding situations. Note that background predicates are removed. In the case of a word co-occurrence class denoting a unary predicate, this intersection contains the correct predicate.

| utterance | situation |
|---|----------------|
| <i>el cuadrado a la izquierda del triangulo</i> | <i>bgsלבgt</i> |
| <i>el triangulo encima del triangulo verde</i> | <i>bgtלבgt</i> |
| <i>el triangulo verde</i> | <i>bgtלבrt</i> |
| <i>el triangulo encima del triangulo azul</i> | <i>bgtלבbt</i> |
| <i>el circulo rojo</i> | <i>brcלבrt</i> |
| <i>el cuadrado</i> | <i>brslbbc</i> |

| class | primary predicates |
|-----------------------------|---------------------------|
| { <i>el</i> } | {} |
| { <i>cuadrado</i> } | { <i>sq, le</i> } |
| { <i>a, la, izquierda</i> } | { <i>gr, sq, tr</i> } |
| { <i>del</i> } | { <i>gr, tr</i> } |
| { <i>triangulo</i> } | { <i>gr, tr</i> } |
| { <i>encima</i> } | { <i>gr, tr, ab</i> } |
| { <i>verde</i> } | { <i>gr, tr, ab</i> } |
| { <i>azul</i> } | { <i>gr, tr, ab, bl</i> } |
| { <i>circulo, rojo</i> } | { <i>re, ci, ab, tr</i> } |

Table 1: Spanish: co-occurrences classes and associated predicates after 6 random examples.

Once a new example is added, co-occurrences classes and associated predicates are updated. Table 2 shows the new word co-occurrence classes (the class {*circulo, rojo*} is split into two different co-occurrence classes) and their associated primary predicates after adding the following example: (*brtlbbt, el triangulo rojo a la izquierda del triangulo azul*). We also show the meaning chosen for each word after receiving these seven input pairs. Note that the algorithm prefers to assign a unary predicate as the meaning of a co-occurrence class if possible. The meanings of *square, triangle, el, a, la* are correct, and also the meaning of *verde* and *azul* (although these two are correct only because of a lucky random choice of a unary predicate from their primary predicates.) All the other words have an incorrect meaning. However, these incorrect and accidentally correct meanings can be guaranteed correct with more examples (as we show in Table 4).

We have shown that this algorithm converges to a function γ' such that $\gamma'(u) = \gamma(u)$ under some further assumptions about the meaning transducer and the input sequence. Here we informally describe the assumptions and state the theorem; for further details of the assumptions, the algorithm and its proof of correctness, please see [2, 3]. Of course, these assumptions cannot be expected to hold in general of natural language; they are intended to give an indication of the limitations of our algorithm.

Assumption 2 is that γ and the word co-occurrence classes of $L(M)$ interact so that at most one word in a co-occurrence class is assigned a predicate, and any word in the class may be assigned the predicate. (This assumption is not satisfied by our Greek transducer: the words *o* and *kyklos* are in the same co-occurrence class, but the output of γ is affected by choosing one or the other to assign the meaning *ci*.) Assumption 3 is that for each word co-occurrence class K of $L(M)$, the set of predicates common to the meanings of all utterances in $L(M)$ containing K is equal to $\gamma(K)$. Assumption 4 is that the input sequence is sufficient to guarantee that the partition

| class | primary predicates | word | meaning chosen |
|-----------------------------|---------------------------|------------------|----------------|
| { <i>el</i> } | {} | <i>el</i> | ε |
| { <i>cuadrado</i> } | { <i>sq, le</i> } | <i>cuadrado</i> | <i>sq</i> |
| { <i>a, la, izquierda</i> } | { <i>le, tr</i> }. | <i>a</i> | ε |
| { <i>del</i> } | { <i>tr</i> } | <i>la</i> | ε |
| { <i>triangulo</i> } | { <i>tr</i> } | <i>izquierda</i> | <i>tr</i> * |
| { <i>encima</i> } | { <i>gr, tr, ab</i> } | <i>del</i> | <i>tr</i> * |
| { <i>verde</i> } | { <i>gr, tr, ab</i> } | <i>triangulo</i> | <i>tr</i> |
| { <i>azul</i> } | { <i>tr, bl</i> } | <i>encima</i> | <i>gr</i> * |
| { <i>circulo</i> } | { <i>re, ci, ab, tr</i> } | <i>verde</i> | <i>gr</i> + |
| { <i>rojo</i> } | { <i>re, tr</i> } | <i>azul</i> | <i>bl</i> + |
| | | <i>circulo</i> | <i>re</i> * |
| | | <i>rojo</i> | <i>tr</i> * |

Table 2: Spanish: co-occurrences classes, associated predicates and meaning chosen after adding one more example. Meanings marked with * are incorrect. Meanings marked with + are lucky guesses.

\mathcal{K}_n converges to the word co-occurrence classes of $L(M)$. Assumption 5 is that for each word co-occurrence class K , the set of non-background predicates occurring in all situations S_i such that K is contained in u_i is equal to the primary versions of the predicates in $\gamma(K)$. Assumption 6 is that if the algorithm assigns the correct values of predicates to word co-occurrence classes whose values are unary predicates, then the input sequence is sufficient to eliminate incorrect non-background binary predicates from $B_n(K)$.

Theorem 1. *Under Assumptions 1 through 6, the learning algorithm finitely converges to a meaning function γ' such that $\gamma'(u) = \gamma(u)$ for every $u \in L(M)$.*

4 Empirical Tests of Learning the Meaning Function

We have implemented and tested our algorithm to learn the meaning function in the example domain of geometric shapes with sets of utterances in a number of natural languages, including Arabic, English, Greek, Hebrew, Hindi, Mandarin, Russian, Spanish, and Turkish. This domain is a simplification of the *Miniature Language Acquisition* task proposed by Feldman et al. [11]. These experiments allow us to assess the robustness of our assumptions for this limited domain and the adequacy of our model to deal with crosslinguistic data. Full details are available in [2].

Although we test our algorithm with limited sublanguages of natural languages, these sublanguages still present several interesting linguistic phenomena that make the learning task nontrivial. For example, depending on the language, the elements of the utterance can have different orders (in Hebrew the adjective comes after the noun, while in English it comes before the noun), there may be grammatical agreement between article and noun, and between noun and adjective (as in Spanish), the object denoted can be placed in first place (as in Arabic and Russian) or in last place (as in Turkish and Hindi), several words together can have just one meaning (for example, in Mandarin *san jiao xin* denotes *tr*), and nouns, articles and adjectives can be declined (e.g., nominative, accusative, genitive in Greek). Even these limited sublanguages present enough linguistic variability to be a challenge for a learning algorithm, particularly because we do not take advantage of morphological information.

4.1 The initial samples

The data for the first set of tests was gathered by asking a native speaker of each language considered to translate the set of fifteen utterances shown in Table 3 with reference to a diagram of the corresponding situation. The respondents provided Romanized spellings of the words to facilitate testing. We then ran the algorithm to learn a meaning function from the resulting utterances paired with the same situations as in the table. In addition, we created a second English sample, labeled Directions, for the same situations in the form of giving “directions” to an object, for example, *go to the red circle and then north to the triangle*.

| utterance | situation |
|---|----------------|
| <i>the triangle</i> | <i>bbt</i> |
| <i>the blue triangle</i> | <i>bbtlbrt</i> |
| <i>the red triangle to the left of the blue square</i> | <i>brtlbbs</i> |
| <i>the circle above the green triangle</i> | <i>bbcabgt</i> |
| <i>the red circle to the right of the green circle</i> | <i>bgclbrc</i> |
| <i>the triangle above the red square</i> | <i>bgtabrs</i> |
| <i>the green triangle</i> | <i>bgtabrs</i> |
| <i>the blue circle</i> | <i>bbcabgt</i> |
| <i>the red triangle to the right of the blue triangle</i> | <i>bbtlbrt</i> |
| <i>the red circle</i> | <i>brc</i> |
| <i>the circle above the square</i> | <i>bbcabrs</i> |
| <i>the circle to the left of the square</i> | <i>bgclbgs</i> |
| <i>the blue circle above the square</i> | <i>bbcabgs</i> |
| <i>the circle to the left of the triangle</i> | <i>bbclbgt</i> |
| <i>the triangle to the right of the circle</i> | <i>bbclbgt</i> |

Table 3: English utterances and situations. Situations are denoted by abbreviations: *bbcabrs* denotes a big blue circle above a big red square

For each language we recorded (1) the translations of the initial sample of fifteen utterances and the corresponding situations, (2) the set of word co-occurrence classes and their associated sets of predicates for this data, and (3) the meaning function chosen on the basis of this data. For each language for which the learning algorithm did not achieve convergence for the initial sample, we also compute the final converged values of (2) and (3), computed as described in Section 4.2.

For the English, Mandarin, Spanish and Directions samples, the fifteen initial examples are sufficient for convergence of the sets of predicates associated with each co-occurrence class of words, and also for the correct resolution of the binary predicates; correct meaning functions are learned in each of these cases. In these cases, there is a single word class associated with each unary predicate; for example, there is a single word class associated with the predicate *tr* and a single word class associated with the predicate *re*. The converged results for Spanish are shown in Table 4.

In the Arabic, Greek, Hebrew, Hindi, Russian and Turkish samples, the fifteen initial examples given are not sufficient to ensure convergence to the final sets of predicates associated with each class of words. For example, in the Hebrew sample, the word classes *hameshulash* and *lameshulash* should both refine to the predicate *tr*, but the sample is only sufficient to refine *lameshulash* to the two predicates *tr* and *bl*. In the construction of the meaning function, one of *tr* and *bl* is randomly chosen for the meaning of *lameshulash*. (Note that because the algorithm does not exploit morphology, *hameshulash* and *lameshulash* must be learned separately.) For the language

| utterance | situation |
|---|----------------|
| <i>el triangulo</i> | <i>bbt</i> |
| <i>el triangulo azul</i> | <i>bbtlbrt</i> |
| <i>el triangulo rojo a la izquierda del cuadrado azul</i> | <i>brtlbbs</i> |
| <i>el circulo encima del triangulo verde</i> | <i>bbcabgt</i> |
| <i>el circulo rojo a la derecha del circulo verde</i> | <i>bgclbrc</i> |
| <i>el triangulo encima del cuadrado rojo</i> | <i>bgtabrs</i> |
| <i>el triangulo verde</i> | <i>bgtabrs</i> |
| <i>el circulo azul</i> | <i>bbcabgt</i> |
| <i>el triangulo rojo a la derecha del triangulo azul</i> | <i>bbtlbrt</i> |
| <i>el circulo rojo</i> | <i>brc</i> |
| <i>el circulo encima del cuadrado</i> | <i>bbcabrs</i> |
| <i>el circulo a la izquierda del cuadrado</i> | <i>bgclbgs</i> |
| <i>el circulo azul encima del cuadrado</i> | <i>bbcabgs</i> |
| <i>el circulo a la izquierda del triangulo</i> | <i>bbclbgt</i> |
| <i>el triangulo a la derecha del circulo</i> | <i>bbclbgt</i> |

| class | primary predicates | word | meaning chosen |
|----------------------|--------------------|------------------|-----------------------|
| { <i>el</i> } | {} | <i>el</i> | ε |
| { <i>triangulo</i> } | { <i>tr</i> } | <i>triangulo</i> | <i>tr</i> |
| { <i>azul</i> } | { <i>bl</i> } | <i>azul</i> | <i>bl</i> |
| { <i>rojo</i> } | { <i>re</i> } | <i>rojo</i> | <i>re</i> |
| { <i>a, la</i> } | { <i>le</i> } | <i>a</i> | ε |
| { <i>izquierda</i> } | { <i>le</i> } | <i>la</i> | ε |
| { <i>del</i> } | {} | <i>izquierda</i> | <i>le</i> |
| { <i>cuadrado</i> } | { <i>sq</i> } | <i>del</i> | ε |
| { <i>circulo</i> } | { <i>ci</i> } | <i>cuadrado</i> | <i>sq</i> |
| { <i>encima</i> } | { <i>ab</i> } | <i>circulo</i> | <i>ci</i> |
| { <i>verde</i> } | { <i>gr</i> } | <i>encima</i> | <i>ab</i> |
| { <i>derecha</i> } | { <i>le</i> } | <i>verde</i> | <i>gr</i> |
| | | <i>derecha</i> | <i>le^t</i> |

Table 4: Spanish: results for the initial sample have converged.

in this group, additional examples of a similar nature are sufficient to ensure convergence to a correct meaning function, as we show in Section 4.2.

There is another kind of accidental association which would require an enlarged domain to remove. For example, in the case of Arabic, the words *alhamraa*, *alkhadraa*, and *alzarkaa* are only used of *aldaerah*, (circle), which means that even after seeing all of the utterances in the restricted sublanguage, the predicates associated with *alhamraa* are both *re* and *ci*, and analogously for *alkhadraa* and *alzarkaa*. A similar phenomenon occurs in the Greek sample; for example, *kokkinos* is associated with both *re* and *ci* even after seeing all the utterances in the restricted sublanguage. In particular, both the Arabic and Greek transducers violate Assumption 3, and correct convergence of the meaning function is not guaranteed. An example run (produced as described in Section 4.2) showing the converged results of the algorithm in the case of Greek is shown in Table 5.

When two objects are mentioned, the denoted object may be mentioned first (as in the Arabic, English, Greek, Hebrew, Spanish, and Russian samples) or second (as in the Directions, Hindi,

| class | primary predicates |
|--------------|--------------------|
| {to} | {} |
| {mple} | {bl} |
| {trigono} | {tr} |
| {kokkino} | {re} |
| {tetragono} | {sq} |
| {sta, tou} | {le} |
| {deksia} | {le} |
| {tetragonou} | {sq, le} |
| {kyklou} | {ci, le} |
| {pano, apo} | {ab} |
| {o, kyklos} | {ci} |
| {aristera} | {le} |
| {kokkinou} | {le, re} |
| {trigonou} | {le, tr} |
| {prasinou} | {le, gr} |
| {kokkinos} | {re, ci} |
| {prasino} | {gr} |
| {ton, kyklo} | {ci, ab} |
| {prasinos} | {gr, ci} |

| word | meaning chosen |
|------------|-----------------|
| to | ε |
| mple | bl |
| trigono | tr |
| kokkino | re |
| tetragono | sq |
| sta | ε |
| deksia | le ^t |
| tou | ε |
| tetragonou | sq |
| kyklou | ci |
| pano | ab |
| apo | ε |
| o | ε |
| kyklos | ci |
| aristera | le |
| kokkinou | re |
| trigonou | tr |
| prasinou | gr |
| kokkinos | ci * |
| prasino | gr |
| ton | ε |
| kyklo | ci |
| prasinos | ci * |

Table 5: Greek: results after convergence; *kokkinos* and *prasinos* are not sufficiently resolved – the chosen meanings are incorrect and marked with *.

Mandarin and Turkish samples.) This affects the binary predicates chosen for the meanings of words. For example, in the English sample *above* is assigned the meaning *ab*, but in the Directions sample, *north* is assigned the meaning *ab^t*, because the second argument of *ab* precedes the first argument of *ab* in the resulting meaning. For example, in *go to the red triangle and then north to the square*, the square is above the triangle, but the second argument of *ab*, namely, the triangle, is mentioned first. A similar consideration applies to left and right (e.g., *east* and *west* in the Directions sample.)

4.2 Randomly generated samples

To explore the issues of whether our theoretical assumptions are satisfied and how many examples may be required to ensure convergence, we constructed meaning transducers for each language in our study and performed a set of experiments using randomly generated samples.

For each language we constructed a meaning transducer capable of expressing the 444 different meanings involving one or two objects.³ There are 12 meanings involving one object: the 3 shape predicates, and the 9 combinations of a color and a shape predicate. There are $432 = 12 \times 3 \times 12$

³For all but Directions example, the transducer was constructed to accept only those 444 utterances; the Directions transducer accepts an infinite language, similarly to the transducer M_1 in Figure 2.

meanings involving two objects related by one of 3 relations: left, right or above. The number of states in the meaning transducer for each language is shown in Table 6.

Given a meaning transducer, we generated a random example as follows. We first randomly generated a situation involving two objects. There are $162 = 9 \times 2 \times 9$ different such situations, because each object is specified by one of three colors (*bl*, *gr*, *re*) and one of three shapes (*tr*, *sq*, *ci*), and there are two possible relations (*le*, *ab*) between them. We then determined all of the utterances accepted by the meaning transducer that are denoting for the selected situation, and selected one of these at random, returning the resulting pair consisting of the situation and denoting utterance.

There are 1476 distinct pairs consisting of a situation and a denoting utterance for that situation. Note that our sampling method does not sample the possible utterances uniformly (because *the square* is a denoting utterance in many more situations than *the blue square to the right of the red triangle*), nor does it sample the situation/utterance pairs uniformly (because some situations have more denoting utterances than others.)

To determine the final co-occurrence classes and their sets of predicates, we ran our algorithm on random samples of 100 or 200 examples, and checked manually whether convergence had occurred. This process has shown that our theoretical assumptions are satisfied and a correct meaning function is found in the following cases: Directions, English, Hebrew, Hindi, Mandarin, Russian, Spanish, and Turkish. For Arabic and Greek, the violations of Assumption 3 noted above mean that a fully correct meaning function is not guaranteed. However, even in these two cases, a largely correct meaning function is achieved.

We then made a set of 10 runs for each language, each run consisting of generating a sequence of random examples and running the algorithm on longer and longer prefixes of it until it reached the final word co-occurrence classes and their sets of predicates for this language. Statistics on the results of the number of examples to convergence of the random runs are shown in Table 6.

| language | correct meanings? | transducer size | median # exs | mean # exs |
|------------|-------------------|-----------------|--------------|------------|
| Arabic | No | 10 | 52.5 | 67.6 |
| Directions | Yes | 8 | 36.0 | 38.3 |
| English | Yes | 11 | 38.5 | 34.9 |
| Greek | No | 20 | 93.0 | 95.9 |
| Hebrew | Yes | 6 | 39.5 | 37.7 |
| Hindi | Yes | 11 | 62.5 | 68.9 |
| Mandarin | Yes | 17 | 37.5 | 40.4 |
| Russian | Yes | 11 | 117.5 | 112.4 |
| Spanish | Yes | 10 | 41.0 | 46.5 |
| Turkish | Yes | 7 | 36.0 | 37.4 |

Table 6: Random test: # examples until convergence in 10 runs

The process generating these statistics is one of waiting until the sampling produces enough variation to eliminate all the incorrect possible associations of each word. As in a coupon collector process, there is a lot of waiting for the last few meanings to refine, because examples to refine them are not very probable. The statistical process is essentially equivalent for the Directions, English, Mandarin and Spanish transducers, yielding a median of about 40 samples. (Differences in their statistics are due to random variation.) Languages with more than one word for each predicate will tend to incur more waiting, for example, Russian, with three versions of each adjective and three of each noun, and Greek, with two forms for triangle and square and three for circle, red and green. Intermediate are Hindi, which has two forms for green and blue, and Arabic, which has two forms

for red, green and blue. Hebrew and Turkish seem not to incur any extra waiting, despite having two forms for each noun – likely because each object mentioned requires a noun but not necessarily an adjective.

Overall, compared with the 162 possible situations, 444 possible utterances, and 1476 possible situation/utterance pairs, a few tens of randomly chosen examples to convergence does not seem excessive, especially as the intermediate results appear to be partially correct.

To get some sense of how this process might scale, we also ran 10 trials with an English transducer with 6 color terms, 6 shape terms, and 4 relation terms (left, right, above and below). The number of situations involving two objects is now 2592 (up from 162) and the number of possible denoting utterances is 7098 (up from 444). For the 10 trials, the mean of the number of examples until convergence was 47.2 and the median was 49.0. This modest increase reflects two contrary tendencies: more terms means more examples to ensure that they are all sampled by a random process, but more terms also means a smaller probability of accidental coincidences in random samples. Comparisons with other work analyzing cross-situational learning [22, 23, 25] may be found in [2].

5 From Telegraphic to Adult Speech

The results in the previous section show that under certain assumptions, a simple algorithm can learn to “comprehend” an adult’s utterance in the sense of producing the same output sequence of predicates, even without mastering the adult’s grammar. In this section we focus on the complementary task of production.

We can model production by the adult as converting a sequence of predicates (p_1, p_2, \dots, p_k) into an utterance $u \in L(M)$ such that $M(u) = (p_1, p_2, \dots, p_k)$, which requires access to the adult meaning transducer. However, to model the child’s production, we assume that the learner does not have access to the adult meaning transducer, and that the child’s production progresses through different stages.

To simplify the child production task, we assume that at the start of the production phase, the learner’s lexicon represents a correct meaning function. Thus, for example, in the learner’s lexicon *blue* is correctly mapped to *bl*, *triangle* to *tr*, *above* to *ab*, *square* to *sq*, and so on. We also assume that the learner has learned to generate correct sequences of predicates (e.g., (bl, tr, ab, sq)), based on the observation that the underlying grammar of sequences of predicates is generally simpler than the grammar of utterances. The child’s initial production strategy can be modeled as generating a correct sequence of predicates and inverting the meaning function to produce a kind of “telegraphic speech.” For example, from (bl, tr, ab, sq) the child may produce *blue triangle above square*.

The adult must be able to comprehend this telegraphic speech (which in our setting is a matter of applying γ without insisting on grammaticality) to arrive at the sequence of predicates intended by the child. At this point, the adult’s production can be used to provide an expansion (e.g., *the blue triangle above the square*).

Our goal in this section is to model how the learner might move from telegraphic speech to speech that is grammatical in the adult’s sense. Moreover, we would like to find a formal framework in which *meaning-preserving corrections* can be given to the child during the intermediate learning stages to study their effect on language learning. We explore the idea of modeling child language production as a machine translation problem, i.e., as the task of translating a sequence of predicate symbols representing the meaning of an utterance into a corresponding utterance in a natural language.

5.1 Subsequential transducers

Subsequential transducers (SSTs) have been widely used in the context of machine learning of translation tasks. SSTs are deterministic finite state models that allow input-output mappings between languages. When an input string is accepted, the SST produces an output string that consists of concatenating the output substrings associated with sequence of edges traversed, together with the substring associated with the last state reached by the input string. Formal definitions can be found in [5, 8].

In [16], it is proved that SSTs are learnable in the limit from a positive presentation of sentence pairs by an efficient algorithm called OSTIA (Onward Subsequential Transducer Inference Algorithm). OSTIA takes as input a finite training set of input-output pairs of sentences, and produces as output an Onward Subsequential Transducer that generalizes the training pairs.

SSTs and OSTIA have been successfully applied to different translation tasks, among others, to translate Spanish sentences describing simple visual scenes to corresponding English and German sentences [7]. They have also been applied to the problem of learning to comprehend language [8, 19]. Moreover, several extensions of OSTIA have been introduced. For example, OSTIA-DR incorporates domain (input) and range (output) models in the learning process, allowing the algorithm to learn SSTs that accept only sentences compatible with the input model and produce only sentences compatible with the output model [18]. Experiments with a simple language understanding task gave better results with OSTIA-DR than with OSTIA [8]. Another extension is DD-OSTIA [17], which instead of considering a lexicographic order to merge states, it uses a heuristic order based on some measure of the equivalence of the states. Results obtained in [17] show that better results can be obtained by using DD-OSTIA in translation tasks from Spanish to English.

6 Empirical Tests of Learning Production

In this section we empirically explore the capabilities of SSTs and OSTIA to model how the learner can move from telegraphic to adult speech. Our experiments were made for the same domain considered in Section 4 (i.e., a limited domain of geometric shapes and their properties and relations). In addition to the nine natural languages used in our previous experiments, we also considered Hungarian.

In order to generate the data sequences for our experiments, we used the meaning transducers constructed in Section 4.2, and we also constructed the corresponding one for Hungarian. We randomly generated 400 non-repeated input-output pairs for each language, each of which contains as input a sequence of predicates (e.g., (bl, tr, ab, sq)) and as output the corresponding utterance in a natural language (e.g., *the blue triangle above the square*). This process was repeated 10 times for each language.

We ran OSTIA on initial segments of each sequence of input/output pairs, of lengths 10, 20, 30, ..., to produce a sequence of subsequential transducers. The whole data sequence was used to test the correctness of each transducer generated during the process: if for some input/output pair (x, y) the transducer translated x to y' , and $y' \neq y$, then it was counted as an *error*. OSTIA was deemed to have *converged* to a correct transducer if all the transducers produced afterwards on the same input/output sequence had the same number of states and edges, and 0 errors on the whole data sequence. Because better results were reported using DD-OSTIA in machine translation tasks, we performed the same experiments using OSTIA and DD-OSTIA.

Figure 3 shows statistics on the number of pairs needed until convergence for OSTIA and DD-OSTIA for all ten languages. We can see that DD-OSTIA requires fewer samples to converge than OSTIA (sometimes dramatically fewer, e.g., Hindi). However, even with DD-OSTIA the number

of samples is in several cases a large fraction of all 444 possible pairs (40% or more are required for Greek, Hindi, Hungarian and Russian). Mandarin needs the smallest number of samples to achieve convergence for both OSTIA and DD-OSTIA. Greek requires the largest number of samples for DD-OSTIA, and one of the largest number of samples for OSTIA.

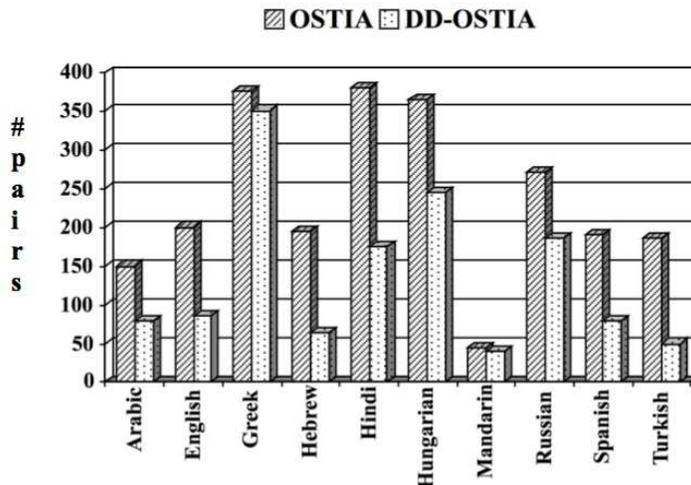


Figure 3: Median number of input-output pairs until convergence in 10 runs needed for OSTIA and DD-OSTIA

Figure 4 shows the mode of the numbers of states of the transducers after convergence learned by OSTIA and DD-OSTIA for each language. Although the transducers produced after convergence by OSTIA and DD-OSTIA correctly translate all the given input/output pairs, the behavior of the transducers may be different on other inputs. Mandarin gives the smallest transducer; one state suffices to translate correct predicate sequences into utterances. In contrast, English and Spanish both require two states to handle articles correctly. Greek gives the largest transducer, consisting of nine states. Despite the evidence of the extremes of Mandarin and Greek, the relation between the size of the transducer for a language and the number of samples required to converge to it is not monotonic, as we can see in Figure 5

These results suggest that OSTIA and DD-OSTIA may be effective methods to learn to translate sequences of predicates into natural language utterances in our domain, given enough examples. However, some of our objectives seem incompatible with the properties of OSTIA. Looking at one of the intermediate transducers produced by OSTIA (see Figure 6), we see that this transducer correctly translates the ten predicate sequences used to construct it, but also produces other translations that are incorrect. For example, the predicate sequence *ci* is translated as *el circulo a la izquierda del circulo verde*, but the learner, using the lexicon, could detect an incompatibility because the lexicon indicates that the translation of this utterance should be *(ci, le, ci, gr)*. Moreover, the intermediate results of the learning process do not seem to have the properties we expect of a learner who is progressing towards mastery of production of a natural language.

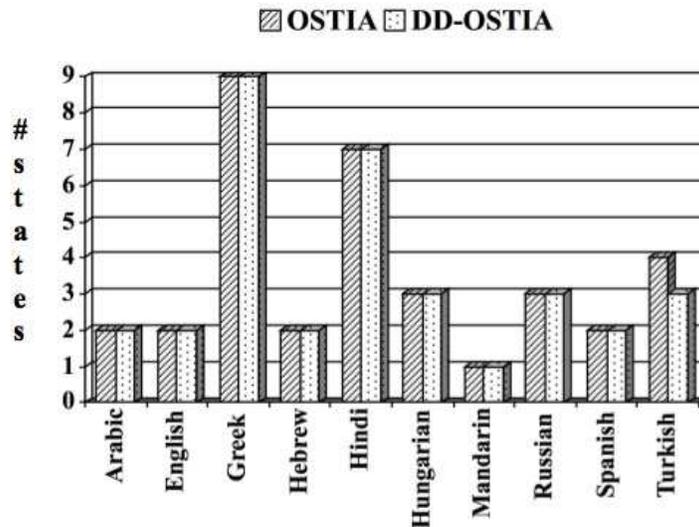


Figure 4: Sizes of transducers at convergence learned by OSTIA and DD-OSTIA

7 Final Remarks and Future Work

In this paper we have presented some experimental results for learning algorithms that use semantics for language comprehension and production. On one hand, it has been shown that under certain assumptions, a simple algorithm can learn to comprehend an adult’s utterance even without mastering the adult’s grammar. On the other hand, we have empirically shown that although we could model the acquisition of language production using SSTs and OSTIA, the framework we consider does not seem suitable for modeling corrections.

One goal in choosing the meaning representation described in this paper was to be able to model an interaction between an adult and child in which the adult provides a meaning-preserving expansion of an utterance of the child. That is, the learner produces a (flawed) utterance that can be understood by the teacher thanks to the shared situation, and then, the teacher responds with a correct utterance for that meaning (i.e., a meaning-preserving correction). The learner, recognizing that the teacher’s utterance has the same meaning but different form, uses the teacher’s utterance, its own utterance and the situation to update its knowledge of the language. Since the properties of OSTIA do not seem to be compatible with our objectives, our next step is to find an appropriate model of how the learner’s language production might evolve in order to be able to model meaning-preserving corrections. Such a model would allow us to explore the roles of semantics and corrections in language learning.

8 Acknowledgments

Preliminary versions of parts of this work were presented in [1, 4]. The authors would like to thank Ronny Dakdouk, Kevin Gold, Melis Inan, Gaja Jarosz, Edo Liberty, José Oncina, Lev Reyzin, Brian Scassellati, Nikhil Srivastava, Antonis Stampoulis, and Yinghua Wu for their help with aspects of this work. The comments of the anonymous reviewers of this paper were very helpful in improving the presentation.

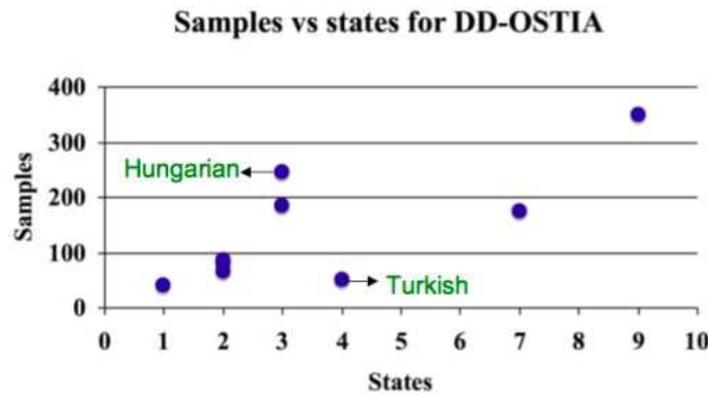
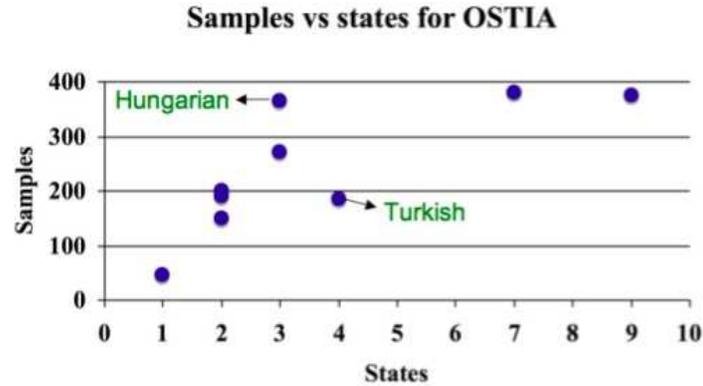


Figure 5: Number of samples versus number of states for OSTIA and DD-OSTIA

References

- [1] D. Angluin and L. Becerra-Bonache. Experiments with an algorithm to learn meaning before syntax. In *ForLing2008*, 2008.
- [2] D. Angluin and L. Becerra-Bonache. Learning meaning before syntax. Technical Report YALE/DCS/TR1407, Computer Science Department, Yale University, 2008.
- [3] D. Angluin and L. Becerra-Bonache. Learning meaning before syntax. In *ICGI*, pages 281–292, 2008.
- [4] D. Angluin and L. Becerra-Bonache. Experiments using OSTIA for a language production task. In *EACL 2009: Workshop on Computational Linguistics Aspects of Grammatical Inference*, pages 16–23, 2009.
- [5] J. Berstel. *Transductions and Context-Free Languages*. PhD thesis, Teubner, Stuttgart, 1979.
- [6] R. Brown and U. Bellugi. Three processes in the child’s acquisition of syntax. *Harvard Educational Review*, 34:133–151, 1964.
- [7] A. Castellanos, I. Galiano, and E. Vidal. Applications of OSTIA to machine translation tasks. In *ICGI*, pages 93–105, 1994.

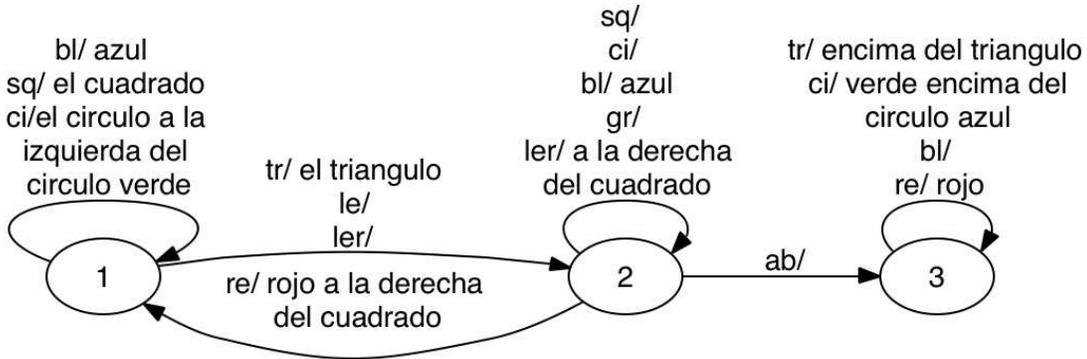


Figure 6: Transducer produced by OSTIA using 10 random unique input-output pairs (predicate sequence, utterance) for Spanish. The predicate le^t is here denoted *ler*.

- [8] A. Castellanos, E. Vidal, and J. Oncina. Language understanding and subsequential transducers. In *ICGI*, pages 11/1–11/10, 1993.
- [9] M.M. Chouinard and E.V. Clark. Adult reformulations of child errors as negative evidence. *Journal of Child Language*, 30:637–669, 2003.
- [10] C. de la Higuera. A bibliographical study of grammatical inference. *Pattern Recognition*, 38:1332–1348, 2005.
- [11] J.A. Feldman, G. Lakoff, A. Stolcke, and S. Hollback Weber. Miniature language acquisition: A touchstone for cognitive science. Technical Report TR-90-009, International Computer Science Institute, Berkeley, California, 1990.
- [12] K. Gold and B. Scassellati. A robot that uses existing vocabulary to infer non-visual word meanings from observation. In *AAAI*, pages 883–888, 2007.
- [13] J. Grimshaw. Form, function, and the language acquisition device. In C.L. Baker and J.J. McCarthy, editors, *The Logical Problem of Language Acquisition*. MIT Press, Cambridge, MA, 1981.
- [14] J. A. Collins Hill. *A Computational Model of Language Acquisition in the Two-year-old*. Indiana University Linguistics Club, Indiana, 1983.
- [15] T. Oates, T. Armstrong, J. Harris, and M. Nejman. On the relationship between lexical semantics and syntax for the inference of context-free grammars. In *AAAI*, pages 431–436, 2004.
- [16] J. Oncina. *Aprendizaje de lenguajes regulares y transducciones subsecuenciales*. PhD thesis, Universitat Politècnica de València, Valencia, Spain, 1991.
- [17] J. Oncina. The data driven approach applied to the OSTIA algorithm. In *ICGI*, pages 50–56, 1998.
- [18] J. Oncina and M.A. Varo. Using domain information during the learning of a subsequential transducer. In *ICGI*, pages 301–312, 1996.

- [19] R. Pieraccini, E. Levin, and E. Vidal. Learning how to understand language. In *EuroSpeech'93*, pages 448–458, 1993.
- [20] S. Pinker. *Language Learnability and Language Development*. Harvard University Press, Cambridge, MA, 1984.
- [21] A.M. Shaerlaekens. *The Two-Word Sentence in Child Language Development: A Study Based on Evidence Provided by Dutch-Speaking Triplets*. Mouton, The Hague, 1973.
- [22] J.M. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61:39–61, 1996.
- [23] K. Smith, A.D.M. Smith, R.A. Blythe, and P. Vogt. Cross-situational learning: a mathematical approach. In P. Vogt, Y. Sugita, E. Tuci, and C. Nehaniv, editors, *Proceedings of Emergence and Evolution of Linguistic Communication III*, pages 31–44. Springer, 2006.
- [24] I. Tellier. Meaning helps learning syntax. In *ICGI*, pages 25–36, 1998.
- [25] Cynthia A. Thompson and Raymond J. Mooney. Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, 18:1–44, 2003.