Yale University
Department of Computer Science

Learning Meaning Before Syntax

Dana Angluin Leonor Becerra-Bonache

YALEU/DCS/TR-1407
July 2008

Learning Meaning Before Syntax

Dana Angluin Leonor Becerra-Bonache

Department of Computer Science, Yale University
P.O.Box 208285, New Haven, CT, USA
{dana.angluin, leonor.becerra-bonache}@yale.edu

Abstract

We present a simple computational model that takes into account semantics for language learning, as motivated by readings in the literature of children's language acquisition and by a desire to incorporate a *robust* notion of semantics in the field of Grammatical Inference. We argue that not only is it more natural to take into account semantics, but also that semantic information can make learning easier, and can give us a better understanding of the relation between positive data and corrections. We propose a model of meaning and denotation using finite-state transducers, motivated by an example domain of geometric shapes and their properties and relations. We give an algorithm to learn a meaning function and prove that it finitely converges to a correct result under a specific set of assumptions about the transducer and examples. We present and analyze the results of empirical tests of our algorithm with natural language samples in the example domain.

1 Introduction

In 1972, Feldman stated that some of the interesting remaining questions in Grammatical Inference were: *inference in the presence of noise, general strategies for interactive presentation and the inference of systems with semantics* [7]. Thirty-six years later, we can state that some of these questions remain still open. In this paper, we are going to focus on the last one: learning systems with semantics.

Results obtained in the field of Grammatical Inference show that learning formal languages from positive data is hard. Gold [9] proves that superfinite classes of languages (i.e., classes containing all finite languages and at least one infinite language) cannot be learned from only positive data, which implies that even the class of regular languages (the smallest one within the Chomsky Hierarchy) cannot be learned from only positive examples. However, learnability results have been obtained by restricting the class of languages to be learned [2, 25], restricting the method for selecting examples [6, 13], providing structural information [15, 17] or making also available negative data [9, 16]. Surveys on this subject can be found in [5, 18].

All these works tend to leave out the semantic information and reduce the learning problem to syntax learning. However, in natural situations, semantic information is also available to the child [1, 8, 11, 12]. Moreover, semantics seems to play an important role in the early stages of children's language acquisition, concretely in the stage known as the two-word stage, in which children go through the production of one word to the combination of two elements. In the view of several authors [19, 20], two-word sentences are "semantic speech", where the context is important to understand their meaning (thanks to the shared context, adult and child can communicate with each other although their grammars are different) and the meanings of the two elements indicate the implied syntactic relations. Later, when the communication is less contextually determined, the child uses more complex syntactic rules and then we can talk about "syntactic speech".

Taking into account that formal language learning is hard and it seems more natural to take into account semantics in language learning, the following question arises: can semantic information simplify the learning problem? Our conjecture is that semantics can make learning easier.

There are several works that take into account semantics for language learning. The ones nearest to our work are [10, 14, 23]. All these approaches are based on Frege's principle of compositionality (this principle states that the meaning of a sentence depends of the meaning of the words involved and of the syntactic rules used to combine them), and use λ-calculus to represent compositional semantics. The input of their algorithms is pairs consisting of a syntactically correct utterance and its meaning. Moreover, these approaches require a correct or nearly correct parse in order to assign a correct meaning to a sentence. However, this assumption fails when we try to analyze two-word sentences because these sentences uttered by the child are not syntactically correct with respect to the adult's grammar.

Inspired by the two-word stage, we propose here a simple computational model that takes into account semantics and does not rely on a complex syntactic mechanism; in that way, we try to represent the fact that, although the child and adult grammars are different, the semantic situation allows communication. Thus, in some sense, our model is more robust with respect to syntax. Moreover, in contrast to other approaches, the input of our learning algorithm is *utterances* and *situations* in which these utterances are produced (like in the two-word stage, where in addition to hearing utterances, children have access to the context in which those utterances are generated).

What kind of utterances should be available to the learner? Ideally we should provide the learner the same kinds of examples that are available to the child. Whereas the availability of positive data (i.e., utterances that are grammatically correct) is generally accepted, the availability of another kind of data, which is often called negative data, remains a matter of significant controversy.

However, there is a kind of information that is specifically available to the child during the two-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 Brown and Bellugi [3]):

CHILD: Eve lunch
ADULT: Eve is having lunch

As we can see, the adult's answer is a grammatical sentence (positive information is given), but constitutes an expansion of an incomplete sentence uttered by the child (in that way, negative information is also obtained, since the expansion suggests that the child's utterance was not grammatically correct). Therefore, a correction is given to the child by means of an expansion. The context will also play here an important role; adult and child share the context, and the adult uses the semantic situation in order to correct the child. Hence, corrections have a semantic component that could facilitate the learning process. Thus, it would be interesting to give an account of that using a formal model.

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. Assuming that the child is in the two-word stage and the context is shared by adult and child, we can imagine two different situations: a situation A, in which the child receives positive information, and looking at the context, he will establish some semantic relations that describes that situation (moreover, sometimes children explicitly repeat the same sentence but using his grammar, i.e., only 2 words, which could be interpreted as an echo of the adult's sentence [4, 12]); a situation B, in which the child produces a sentence and receives an expansion of it (then, the child can see whether the adult misunderstands his message, and also, how to express the same meaning using the adult's grammar).

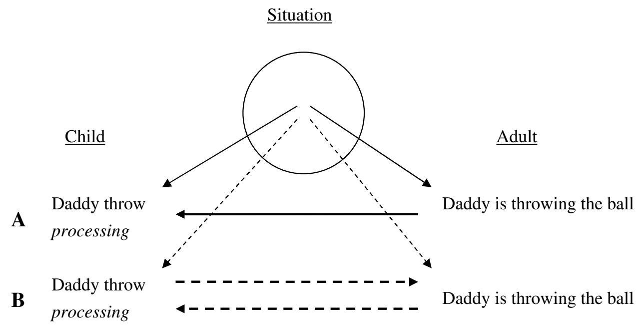

Figure 1: Adult and child share the same situation. In case A, child receives positive data. In B, a correction is returned to the child.

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 correspondence adult's 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.

Therefore, based on ideas from linguistics, cognitive science and computer science, we propose a new computational learning model to take into account the context, semantics, positive data and corrections. As a first step in the formulation of such a model, we focus on a simple formal framework. The model should accommodate at least two different tasks: comprehension and production. However, in this paper, we focus only on the comprehension task. The scenario we consider is cross-situational and supervised, i.e. the teacher provides to the learner several examples pairs consisting of a situation and a utterance that denotes something in the situation. The goal of the learner is to learn the meaning function, allowing the learner to comprehend novel utterances.

The remainder of the paper is organized as follows. First, we describe the meaning and denotation functions used by the teacher to provide examples to the learner (Section 2). Then, we examine simple strategies the learner may use to try to learn the meaning and denotation functions of the teacher (Section 3). Based on all these ideas, we present an algorithm to learn a meaning function, and we prove its correctness (Section 4). We present and analyze the results of tests of our algorithm with samples of several natural languages in a restricted domain (Section 5). We conclude with a discussion of the computational feasibility of the algorithms used by the teacher and the learner, the implications of our approach and future work (Section 6).

2 A Model of Meaning and Denotation

In this section we specify a class of very simple meaning and denotation functions which are used by the teacher to provide examples for the learner.

2.1 Meaning 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 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 occuring 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.

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*, *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. The transition function is partially defined in Figure 2. Undefined transitions go to a non-final dead state, q_7.

$L(M_1)$ contains such utterances 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.* [1] 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 $< bl, tr, ab, sq >$.

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

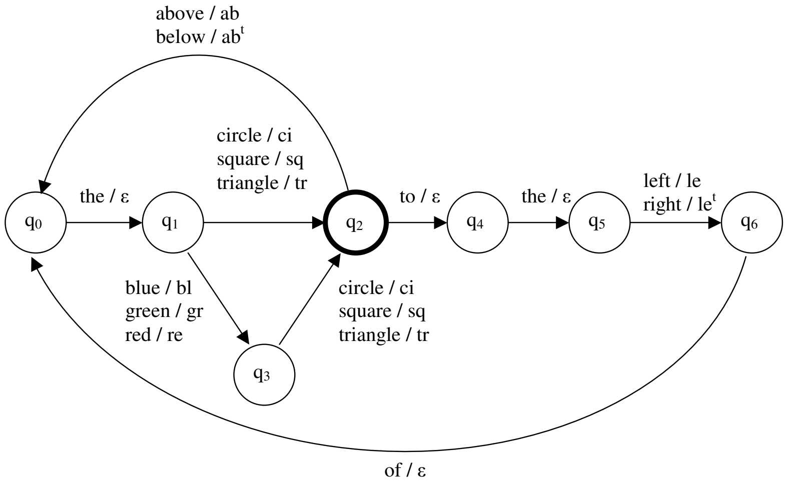

Figure 2: Meaning transducer M_1

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 $x_1, x_2, \ldots$ be distinct variables and $t_1, t_2, \ldots$ 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. Working left to right, 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

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

The definition of π reflects a strong restriction on the way properties and relations can be expressed by a meaning transducer.

2.3 Situations and denotation functions

Next we define situations, which represent the objects, properties and binary relations that are noticed in some environment of the teacher or learner. These are not meant to be exhaustive representations of the environment, but the results of sensing and interpreting part of the environment. A **situation** is a finite set of primary ground atoms. Note that 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. Thus,

$$things(S_1) = \{t_1, t_2\}$$

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

To determine the denotation of an utterance u in a situation S, the teacher takes the meaning $\pi(M(u))$ of u and attempts to match it to a subset of 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 $V = \{x_1, \ldots, x_k\}$ denote the variables that occur in $\pi(M(u))$ and T denote the things in the situation S. A **match** of $\pi(M(u))$ to 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 $\pi(M(u))$ to 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 Strategies for Learning Meanings

Next we examine simple strategies the learner may use to try to learn (approximations of) the meaning and denotation functions of the teacher. For the meaning function, we assume that the learner receives a sequence of pairs (S_i, u_i) from the teacher, where S_i is a situation, and u_i is an utterance with a denotation in the situation S_i. For the denotation function, we assume that the learner receives triples (S_i, u_i, d_i) where S_i is a situation, u_i is a denoting utterance in S_i, and d_i indicates which thing, t_j, in the situation is denoted by u_i. In practice, the denoted object might be indicated by non-linguistic means, e.g., pointing at it. This setting gives rather less information than pairs consisting of an utterance $u \in L(M)$ and its meaning $\pi(M(u))$. Learning the denotation function is just a question of setting the parameter *which* correctly if the meaning function has been learned; we focus on the latter task.

In this definition we are assuming that the learner and teacher share the relevant situation S_i. This is a strong assumption, which may not be satisfied in practice if the learner and teacher are paying attention to very different aspects of their shared environment.

Given a pair (S, u) of situation and utterance, the learner knows that the teacher's transducer M may map some words in u to the empty result, ε, but each other word in u must have been mapped by γ either to one of the predicates in S, or to r^t, where r is a binary predicate in S.

In general, the mapping of a word by M depends on the state that M is in when the word is encountered. However, in the transducer M_1, the output map is state-independent, at least for states other than the dead state. To simplify the learning problem, in this paper we make the following **state-independence assumption**.

Assumption 1. *For all states $q \in Q$ and words $w \in W$, $\gamma(q, w)$ is independent of q.*

Thus we write $\gamma(w)$ instead of $\gamma(q, w)$. Under this assumption, knowing γ is sufficient to compute $M(u)$ for any $u \in L(M)$; we apply γ sequentially to the words of u and form the sequence of results. And because the path-mapping function π is fixed, knowing γ is sufficient to compute the meaning $\pi(M(u))$ of any $u \in L(M)$. In this case we may think of M as separated into a finite state acceptor for $L(M)$ and the function γ to compute the outputs for elements of $L(M)$.

3.1 A cross-situational conjunctive strategy

Given the state-independence assumption, we consider a cross-situational conjunctive learning strategy. Cross-situational learning has been investigated by [21,22,24], among others. For each encountered word w, we consider all utterances u_i containing w and their corresponding situations S_i, and form the intersection of the sets of predicates occurring in these S_i. That is, for each encountered word w let

$$C(w) = \bigcap \{predicates(S_i) : w \in c(u_i)\}.$$

Because u_i is a correct denotation in S_i, if $\gamma(w)$ is a primary predicate, that predicate must be in $C(w)$. Similarly, if $\gamma(w) = r^t$, then r must be in $C(w)$. Hence, if $C(w)$ is empty then the learner may correctly conclude that $\gamma(w) = \varepsilon$.

Continuing with our earlier example, suppose we apply this approach to the pairs of utterances and situations shown in the left part of Table 1. Each situation is described by an abbreviation: for example *brtlbbs* represents the situation of a big red triangle to the left of a big blue square.

For the data in the left part of Table 1, observe that every $C(w)$ contains the predicate *bi* because it is present in every situation. Such a predicate is a **background predicate**. Let $C'(w)$

utterance	situation
the triangle	*bbt*
the blue triangle	*bbtlbrt*
the red triangle to the left of the blue square	*brtlbbs*
the circle above the green triangle	*bbcabgt*
the red circle to the right of the green circle	*bgclbrc*
the triangle above the red square	*bgtabrs*
the green triangle	*bgtabrs*
the blue circle	*bbcabgt*
the red triangle to the right of the blue triangle	*bbtlbrt*
the red circle	*brc*
the circle above the square	*bbcabrs*
the circle to the left of the square	*bgclbgs*
the blue circle above the square	*bbcabgs*
the circle to the left of the triangle	*bbclbgt*
the triangle to the right of the circle	*bbclbgt*

w	$C'(w)$
the	$\emptyset$
triangle	$\{tr\}$
circle	$\{ci\}$
square	$\{sq\}$
blue	$\{bl\}$
red	$\{re\}$
green	$\{gr\}$
above	$\{ab\}$
left	$\{le\}$
right	$\{le\}$
to	$\{le\}$
of	$\{le\}$

Table 1: English utterances and situations; values of $C'(w)$

to be $C(w)$ with all background predicates removed. The values of $C'(w)$ for this data are shown in the right part of Table 1. There is no entry for the word *below* because it is not encountered in the examples.

The results in Table 1 for the words *the*, *triangle*, *circle*, *square*, *blue*, *red*, *green*, *above* and *left* agree with the values of the output function for M_1, but the values for *right*, *to* and *of* disagree. In the case of *right*, the difficulty is that the value should be le^t instead of le. This arises because although derived predicates may be meanings, they do not occur in situations. Additional processing is necessary to get the order of arguments correct for binary predicates. As an example of the kind of information we may use, to determine that ab rather than ab^t is the correct image of the word *above*, we examine the pair consisting of the utterance *the circle above the green triangle* and the situation *bbcabgt*, that is

$$\{bi(t_1), bl(t_1), ci(t_1), ab(t_1, t_2), bi(t_2), gr(t_2), tr(t_2)\}.$$

With the C' values learned for *circle*, *green* and *triangle*, the choice of ab for *above* leads to a match for this utterance, because $ab(t_1, t_2)$ is supported by the situation, while $ab^t(t_1, t_2)$ is not. This approach relies on the correctness of the object identifications as established by the unary predicates.

For the words *to* and *of*, their occurrence in the phrases *to the left of* and *to the right of* ensure that when they occur, the binary predicate le will be present in the situation. However, for these words, the attempt to assign a definite order of arguments to the binary predicate le will fail, because they occur with both argument orders. Thus, solving the problem of argument orders will resolve these meanings as well.

3.2 Other languages, other phenomena

We have gathered comparable examples for several other languages, which exhibit various other phenomena. For example, in our sample for Mandarin, a circle is designated as *yuan* or *yuan xing*, a triangle as *san jiao xing* and a square as *zheng fan xing*. The English utterance *the triangle below the circle* is rendered as *yuan xing xia mian de san jiao xing*. In this case, the denotation of the utterance (the triangle) is mentioned last in the utterance rather than first; this is a case in which the parameter *which* must be *last* rather than *first*.

Also in our sample for Mandarin, *san* and *jiao* always co-occur, and both of their C' values are the unary predicate tr; analogously, *zheng* and *fan* both have the value sq. If two (or more) words always co-occur and have a non-empty meaning, there may be no evidence for which word should be assigned the meaning. In our sample, tr can be assigned to either *san* or *jiao* and the denotation function will be unaffected. A better general solution might be to recognize lexical items that are combinations of words.

This co-occurrence phenomenon also occurs in our sample for Greek: *o* and *kyklos* always co-occur, though not always contiguously, and both have the C' value ci, though in fact *o* is an article. Another phenomenon is present in our sample for Greek: the word for circle appears in three forms: *kyklos*, *kyklou* and *kyklo*, depending on whether it is the object of a preposition, and, if so, which preposition. Examples of this are: *o kyklos pano apo to tetragono* (*the circle above the square*), *to trigono sta deksia tou kyklou* (*the triangle to the right of the circle*), and *to tetragono pano apo ton prasino kyklo* (*the square above the green circle*).

A combination of morphological and semantic evidence would suggest that these three words are in fact one word. If, however, we treat them as separate words, in the case of *kyklou*, the binary predicate le will be present in every situation in which the word is used, so that its C' value is $\{ci, le\}$ in the limit. Similarly, *kyklo* will always co-occur with the binary predicate ab. In our model, each word maps to at most one predicate, so we would like a criterion to select one of the two possibilities.

4 The Learning Algorithm

Based on the ideas discussed above, we propose a learning algorithm, and give set of assumptions under which it finitely converges to a correct meaning function.

4.1 Further assumptions about M

We make additional assumptions about the meaning transducer M. In Assumption 1, we have already assumed that the output function $\gamma(w)$ depends only on the input word w. If W' is a set of words, we define

$$\gamma(W') = \{\gamma(w) : w \in W', \gamma(w) \neq \varepsilon\}.$$

We define the set of all utterances in $L(M)$ that contain w:

$$L_M(w) = \{u \in L(M) : w \in c(u)\}.$$

A word w_1 **implies** a word w_2, denoted $w_1 \rightarrow_M w_2$ if $L_M(w_1) \subseteq L_M(w_2)$; this is true if every utterance in $L(M)$ that contains w_1 also contains w_2. Two words w_1 and w_2 **always co-occur**, denoted $w_1 \leftrightarrow_M w_2$, if $L_M(w_2) = L_M(w_1)$; that is, they occur in exactly the same set of utterances in $L(M)$. The relation of co-occurrence is an equivalence relation; its equivalence classes are **co-occurrence classes**. In our English example, the words *to* and *of* always co-occur and each one is implied by the word *left*. In our Mandarin example, the words *san* and *jiao* always co-occur.

To deal with co-occurrence classes instead of words as the units to which meanings are assigned, we assume that γ is well-behaved with respect to co-occurrence classes. We say that γ is **single-valued** if for every co-occurrence class K, γ assigns a nonempty output to at most one word from K.

Assumption 2. *The output function γ is single-valued and for any single-valued output function γ' such that $\gamma(K) = \gamma'(K)$ for all co-occurrence classes K, $M(u) = M'(u)$ for every $u \in L(M)$, where M' is M with output function γ'.*

For example, in our sample of Mandarin: either *san* or *jiao* can be assigned the meaning *tr* without affecting the resulting values $M(u)$ for utterances $u \in L(M)$. This assumption is not true in our Greek example: if *o* rather than *kyklos* is assigned the output ci, then the output of *o mple kyklos pano apo to tetragono* is changed from $< bl, ci, ab, sq >$ to $< ci, bl, ab, sq >$. (As this does not affect the denotation function, perhaps this assumption should be weakened.)

We next assume that the language of denoting utterances and their meanings are sufficient to determine the value of γ for each co-occurrence class. Define for each co-occurrence class K,

$$P_M(K) = \bigcap\{c(M(u)) : u \in L(M), K \subseteq c(u)\}.$$

This is all predicates common to meanings of utterances from $L(M)$ that contain the words in K. Note that for all co-occurrence classes, $\gamma(K) \subseteq P_M(K)$. The following assumption strengthens this to equality.

Assumption 3. *For all co-occurrence classes K, $\gamma(K) = P_M(K)$.*

This assumption holds of the transducer M_1. For example, the value of P_M for the co-occurrence class $\{of, to\}$, is $\emptyset$, witnessed by the utterances *the circle to the right of the square*, *the triangle to the left of the circle* and *the square to the right of the triangle* and their corresponding sets of predicates: $\{ci, le^t, sq\}$, $\{tr, le, ci\}$ and $\{sq, le^t, tr\}$. In the case of Greek there are occurrences of *kyklou* with both *sta deksia* and *sta aristera* that eliminate both le and le^t from the value of P_M.

Lemma 1. *Under Assumptions 1, 2, and 3, knowledge of the set of co-occurrence classes K and the values $P_M(K)$ is sufficient to compute the value of $\gamma(u)$ for every $u \in L(M)$.*

Proof. Define γ' as follows. For each co-occurrence class K such that $P_M(K)$ is nonempty, select one word $w \in K$ and set $\gamma'(w) = p$, where p is any element of $P_M(K)$, and set $\gamma'(w') = \varepsilon$ for all other $w' \in K$. If $P_M(K) = \emptyset$, then set $\gamma'(w) = \varepsilon$ for all $w \in K$.

By Assumption 2, $\gamma(K)$ is single-valued and by Assumption 3, $P_M(K) = \gamma(K)$. Thus $\gamma'(K) = \gamma(K)$ for all co-occurrence classes K, and γ' is single-valued by definition. Hence by Assumption 2, $\gamma'(u) = \gamma(u)$ for every $u \in L(M)$. □

However, in the setting we consider, what is observed is the primary versions of binary predicates. We therefore define a variant of $P_M(K)$ in which predicates are first transformed to their primary versions.

$$PP_M(K) = \bigcap\{primary(c(M(u))) : u \in L(M), K \subseteq c(u)\}.$$

For the example of the transducer M_1 and the co-occurrence class $\{of, to\}$, the value of PP_M is $\{le\}$ because whenever these two words occur in an utterance u from $L(M)$, either *right* or *left* occurs, and therefore *le* occurs in $primary(c(M(u)))$. Similarly, in the case of Greek, the value of PP_M for the class containing *kyklou* consists of *ci* and *le*.

A useful property of $PP_M(K)$ is that it gives correct information about the unary predicates.

Lemma 2. *For every co-occurrence class K, $PP_M(K) \cap P = P_M(K) \cap P$.*

Proof. Let $p \in P$. Then $primary(p) = p$. If $p \in PP_M(K)$, then for every utterance $u \in L(M)$ such that $K \subseteq c(u)$, $p \in primary(c(M(u))$, thus $p \in c(M(u))$ and therefore $p \in P_M(K)$. Conversely, if $p \in P_M(K)$ then for every utterance $u \in L(M)$ such that $K \subseteq c(u)$, $p \in c(M(u))$, thus $p \in primary(c(M(u))$ and therefore $p \in PP_M(K)$. □

4.2 The learning algorithm

We assume that the learning algorithm receives examples (S_i, u_i) for $i = 1, 2, \ldots$ and responds to each one by hypothesizing a meaning function γ_n based on the first n examples. The criterion of success is whether the algorithm finitely converges to a meaning function γ' such that $\gamma(u) = \gamma'(u)$ for all utterances $u \in L(M)$.

After receiving the example (S_n, u_n), the learner computes the intersection of the sets of predicates seen in every situation so far, as follows.

$$G_n = \bigcap_{i=1}^{n} predicates(S_i).$$

Let G be the set of **background predicates**, that is, all predicates that occur in every situation S_i.

$$G = \bigcap_{i} predicates(S_i).$$

Then G_n finitely converges to G, because the set of predicates in any situation is finite.

The algorithm maintains a partition $\mathcal{K}_n$ of the words it has seen, in which two words w_1 and w_2 are in the same class if they occur in exactly the same set of utterances u_i with $1 \leq i \leq n$. For each class $K \in \mathcal{K}_n$, the learning algorithm computes the set of unary predicates that occur in every situation S_i for which the utterance u_i contains the class K:

$$U_n(K) = P \cap \bigcap \{predicates(S_i) : 1 \leq i \leq n, K \subseteq c(u_i)\}.$$

The algorithm uses these sets to define a **partial meaning function** g_n as follows. For each class $K \in \mathcal{K}_n$, if $(U_n(K) - G_n)$ is nonempty then the algorithm selects one word $w \in K$ and one predicate $p \in (U_n(K) - G_n)$ and defines $g_n(w) = p$. For all other words, the algorithm defines $g_n(w) = \varepsilon$.

The map g_n translates any utterance into a sequence of unary predicates. For example, using the map g_n derived from the data in Table 1, the translation of *the green circle to the right of the red triangle* is $< gr, ci, re, tr >$.

Resolving argument order. The partial meaning function g_n is used to try to gather information about the possible orders of arguments of binary predicates as follows. Let u be a denoting utterance in a situation S, with partial translation $g_n(u) =< p_1, p_2, \ldots, p_k >$. Let $< t_{i_1}, t_{i_2}, \ldots, t_{i_r} >$

be a finite sequence of distinct things from the situation S. We say that this sequence is **compatible** with the partial translation $g_n(u)$ if there exists a partition of the sequence $< 1, 2, \ldots, k >$ into r (possibly empty) non-overlapping consecutive intervals $I_1, I_2, \ldots, I_r$ such that $p_\ell(t_{i_j})$ is supported by S for every $j = 1, \ldots, r$ and $\ell \in I_j$.

We define the set of possible binary predicates $possible(S, u)$ as follows. For each atom $r(t_i, t_j)$ in S, r is included in $possible(S, u)$ if there is an ordering compatible with $g(u)$ in which t_i immediately precedes t_j, and r^t is included in $possible(S, u)$ if there is an ordering compatible with $g(u)$ in which t_j immediately precedes t_i. Note that

$$primary(possible(S, u)) \subseteq R \cap predicates(S).$$

For example, if the situation S is a big red triangle (t_1) to the left of a big green circle (t_2), with a big red square (t_3) below the circle, then the only orderings compatible with $< gr, ci, re, tr >$ are $< t_2, t_1 >$, $< t_3, t_2, t_1 >$, $< t_2, t_3, t_1 >$ and $< t_2, t_1, t_3 >$. Then in the computation of $possible(S, u)$, le^t is included (because $le(t_1, t_2)$ is in the situation and t_2 immediately precedes t_1 in some compatible ordering) but le is not (because t_1 does not immediately precede t_2 in any compatible ordering.) Considering the occurrences of t_2 and t_3 in the compatible orderings, both ab and ab^t will be included.

The learner defines for each class $K \in \mathcal{K}_n$ a set of binary predicates as follows.

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

Finally, the learner uses $B_n(K)$ to extend g_n to a hypothesized meaning function γ_n as follows. For each w such that $g_n(w) \neq \varepsilon$, $\gamma_n(w)$ is set to $g_n(w)$. For each class $K \in \mathcal{K}_n$, if $(U_n(K) - G_n) = \emptyset$ and $(B_n(K) - G_n) \neq \emptyset$, then a word w is selected from K and a predicate q from $(B_n(K) - G_n)$ and $\gamma_n(w)$ is set to q. For all remaining words w, $\gamma_n(w)$ is set to ε. This concludes the description of the learning algorithm.

We note that the algorithm prefers to assign a unary predicate as the meaning of a co-occurrence class K if possible. This means that it prefers ci to le as the meaning of the co-occurrence class of *kyklou* in our Greek example.

4.3 Correctness of the algorithm

Our goal in this section is to prove the following theorem. To do so, we first make some additional assumptions about the sequence of examples (S_i, u_i).

Theorem 3. *Under Assumptions 1 through 6, the learning algorithm finitely converges to a meaning function γ' such that $\gamma'(u) = \gamma(u)$ for every $u \in L(M)$.*

Our first assumption about the data sequence (S_i, u_i) guarantees that the partition $\mathcal{K}_n$ finitely converges to the correct co-occurrence classes for $L(M)$. For the following Assumptions, we consider only words w and co-occurrence classes K that actually appear in some example (S_i, u_i).

Assumption 4. *For all pairs of words w_1 and w_2, $w_1 \leftrightarrow_M w_2$ if and only if for all i, $c(u_i)$ contains both w_1 and w_2 or neither w_1 nor w_2.*

This assumption is satisfied for our English example by the data in Table 1. The second assumption about the sequence of examples allows the algorithm to compute $PP_M(K)$ in the limit.

If K is a co-occurrence class, let

$$C(K) = \bigcap_i \{predicates(S_i) : K \subseteq c(u_i)\},$$

and

$$C'(K) = C(K) - G.$$

Assumption 5. *For each co-occurrence class K, $C'(K) = PP_M(K)$.*

Note that this implies that no background predicate is in $\gamma(K)$ for a co-occurrence class K that appears in some example (S_i, u_i). This assumption is satisfied for our English example by the data in Table 1. Our final assumption regarding the data is that if the unary predicates are learned correctly, then compatibility considerations are enough to rule out any incorrect binary predicates.

Assumption 6. *Suppose g is a partial meaning function such that for all co-occurrence classes K, $g(K) = \gamma(K)$ if $\gamma(K)$ is a unary predicate and $g(K) = \emptyset$ otherwise. Then for every co-occurrence class K such that $\gamma(K)$ is not a unary predicate and every predicate $q \in (R \cup R^t - G)$, such that $q \notin \gamma(K)$, there exists an example (S_i, u_i) such that $q \notin possible(S_i, u_i)$, where possible is computed with respect to g.*

This assumption is satisfied by the data in Table 1.

Proof of Theorem 3. By Assumption 4, the partition $\mathcal{K}_n$ finitely converges to the correct co-occurrence classes of $L(M)$, so let n be sufficiently large that this is true. Because G_n finitely converges to the background predicates G and $U_n(K)$ finitely converges to the unary predicates in $PP_M(K)$, we have that $(U_n(K) - G_n)$ finitely converges to the unary predicates in $PP_M(K)$, which by Lemma 2 and Assumption 3 are just the unary predicates in $\gamma(K)$. If n is also sufficiently large that this is true, we must have $g_n(K) = \gamma(K)$ for all co-occurence classes K such that $\gamma(K)$ is a unary predicate and $g_n(K) = \emptyset$ for all other co-occurrence classes K.

Thus g_n satisfies the hypotheses of Assumption 6. Consider any co-occurrence class K such that $\gamma(K)$ is not a unary predicate. If $\gamma(K) = \emptyset$, then for every binary predicate $q \in (R \cup R^t - G)$, there exists an example (S_i, u_i) such that $q \notin possible(S_i, u_i)$. Thus, for n sufficiently large, $(B_n(K) - G_n) = \emptyset$ and $\gamma_n(K) = \emptyset$.

Suppose $\gamma(K) = \{q\}$ for some $q \in R \cup R^t$. Then $q \notin G$, so $q \in (R \cup R^t - G)$, by Assumption 5. For all sufficiently large n, q is in $possible(S_i)$ if $K \subseteq c(u_i)$ for all $1 \leq i \leq n$. This is true because u_i is a denoting utterance in S_i, so there is a match f from $x_1, \ldots, x_k$ to things in S_i such that all the atoms of $\pi(M(u_i))$ are supported in S_i. Thus, there is an ordering $f(x_1), f(x_2), \ldots, f(x_k)$ of things from S_i compatible with the subsequence of $M(u_i)$ consisting of unary predicates (which is equal to $g_n(u_i)$), and in which $q(t_j, t_{j+1})$ for some j. Thus, $q \in (B_n(K) - G_n)$ for all sufficiently large n. Every other predicate from $(R \cup R^t - G)$ is eliminated by some example (S_i, u_i), by Assumption 6.

Thus, for sufficiently large n, γ_n finitely converges to a meaning function γ' such that $\gamma'(K) = \gamma(K)$ for all co-occurrence classes K of $L(M)$. By Assumption 2, $\gamma'(u) = \gamma(u)$ for all utterances $u \in L(M)$. $\square$

5 Testing on Natural Language Samples

We have implemented and tested our algorithm 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. These experiments allow us to assess the robustness of our assumptions for this domain and the adequacy of our model to deal with crosslinguistic data.

5.1 The initial samples

The first set of tests consisted of asking native speakers to translate the set of fifteen utterances shown in Table 1, and then running the algorithm to learn a meaning function from the resulting utterances paired with the same situations as in the table. The respondents provided Romanized spellings of the words to facilitate testing. In addition, we created a second English sample 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.*

Appendix A gives results for each language: (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 initial sample does not achieve convergence, we also give final converged values of (2) and (3), computed as described in Section 5.2.

For the English, Mandarin, Spanish and English 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 *tr* and a single word class associated with *re*.

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 predicate *tr* and *bl*. In the construction of the meaning function, *tr* is randomly chosen for the meaning of *lameshulash* in the particular run shown. This kind of accidental association can be removed with more examples of a similar nature, as we show in Section 5.2.

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 convergence has occurred, 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 convergence.

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 English Directions, Hindi, Mandarin and Turkish samples.) This affects the binary predicates chosen for the meanings of words. Thus, in the English sample *above* is assigned the meaning *ab*, but in the English 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 (see *east* and *west* in the English Directions sample.)

5.2 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. There are 12 meanings involving one object: the 3 shape predicates, and the 9 combinations of a color and a shape predicate. There are $432 = 12 \times 3 \times 12$ 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 2.

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 large random samples (100 or 200 examples) and checked manually whether convergence had occurred; the results of one such run are shown for those languages for which the initial sample did not achieve convergence. 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 issues noted above regarding adjectives with the words for circle mean that our assumptions are violated, and 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 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 2.

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

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 2: Results of random tests: examples until convergence in 10 runs

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 extravagant, especially as the intermediate results appear to be partially correct.

To get some sense of how this process might scale, we ran 10 analogous 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.

Siskind [21] presents a formalization of cross-situational lexical acquisition and a constraint-satisfaction method that successfully solves very large instances of it. In his experimental setup he uses synthetic grammars resembling English or Japanese, and presents examples each consisting of an utterance and its meaning together with 3 randomly generated uncorrelated distractor meanings. The number of utterances in his samples seems to scale linearly in the number of words in the lexicon. Thompson and Mooney [24] formalize the problem of learning a covering lexicon given a set of examples consisting of an utterance and its meaning represented as a rooted labelled tree; their formalization permits lexical ambiguity. Their system compares favorably with Siskind's in a learning a natural language interface to a geographic database. They remark on the lack of large corpora annotated with semantic representations and advocate active learning to reduce the annotation burden. Smith et al. [22] present a mathematical analysis and experimental results for a simplified model of cross-situational learning of a lexicon. In their model of whole language learning, there are M meanings and each example consists of an utterance and C possible meanings for the utterance, with the correct meaning and $C - 1$ randomly chosen distractor meanings, which are not equal to the target meaning but are otherwise uncorrelated with it. They give an asymptotic analysis of the case when C is much smaller than M, in which case approximately $2M \ln M$ examples will be required. This can be interpreted in terms of the coupon collector problem as approximately the expected number of examples to sample each meaning at least twice; in the first sample the utterance acquires a confounded meaning, which refines to the correct meaning with the second example (because it is unlikely that any incorrect meaning will be repeated.)

In our setting, if we view the all the incorrect possible denoting utterances in a situation as distractor meanings, there are typically more than 3, and they are correlated with the correct meaning by involving some of the same predicates and relations. Because of compositionality, the number of examples required for convergence scales much more slowly than $M \ln M$, where M is the total number of meanings of utterances, contrary to the model of Smith et al. Using a coupon collector argument, it must scale at least as $L \ln L$, where L is the number of items in the target lexicon, because convergence requires each lexical item to be sampled at least once. A more refined analysis of the number of examples required in our setting is desirable.

6 Discussion and Future Work

Another relevant issue is the computational feasibility of the algorithms used by the teacher and learner. An important parameter is the number of things in the example situations; some of our methods do not appear to scale polynomially in this parameter. For example, consider the problem of determining whether there is a match of $\pi(M(u))$ in a situation S. If there are N variables and at least N things, the problem includes as a special case finding a directed path of length N in the situation graph, which is NP-hard in general. Also, the method of determining the order of arguments of binary predicates potentially involves considering all possible orderings of the distinct things in a situation. However, it seems unlikely that human learners cope well with situations involving arbitrarily many things, and it therefore seems important to find good models of focus of attention.

The child's comprehension in the two-word stage is limited to utterances that refers to something present in that moment. Hence, the situation in which these utterances are produced is used by child and adult to understand the meaning of each other's utterances. We have argued that if the situation plays this important role in the communication between child and adult, then learning from positive data and learning from corrections may not be so different; in both cases, the adult version of the utterance is provided to the child and also their common meaning. Our model begins to suggest how this may be possible.

Moreover, our model suggests that learning meaning not only facilitates learning syntax, but also precedes it. Therefore, we agree with Tellier's suggestion [23] that "the acquisition of a conceptual representation of the world is necessary *before* the acquisition of the syntax of a natural language can start."

Further work is required to relax some of the more restrictive assumptions. For example, the limitations of the current framework mean that disjunctive meanings (for example, a color term that applies to both blue and green things) cannot be learned, nor can a function that assigns meanings to more than one of a set of co-occurring words. Statistical approaches may produce more powerful versions of the models we consider. We plan to continue to develop our model to incorporate production tasks for the learner, as well as corrections or expansions from the teacher. Our goal is to contribute to a deeper theoretical understanding of the role of semantics in language acquisition.

7 Acknowledgments

The work of the second author was supported by a Marie Curie International Fellowship within the 6^{th} European Community Framework Programme. The authors would like to thank Ronny Dakdouk, Kevin Gold, Melis Inan, Gaja Jarosz, Edo Liberty, Lev Reyzin, Brian Scassellati, Nikhil Srivastava, Antonis Stampoulis, and Yinghua Wu for their help with aspects of this work.

References

[1] J.R. Anderson. Induction of augmented transition networks. *Cognitive Science*, 1:125–157, 1977.

[2] D. Angluin. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29(3):741–765, 1982.

[3] R. Brown and U. Bellugi. Three processes in the child's acquisition of syntax. *Harvard Educational Review*, 34:133–151, 1964.

[4] R. Brown and C. Fraser. The acquisition of syntax. In C.N. Cofer and B.S. Musgrave, editors, *Verbal behavior and learning: Problems and processes*, pages 158–197. McGraw-Hill, New York, 1963.

[5] C. de la Higuera. A bibliographical study of grammatical inference. *Pattern Recognition*, 38:1332–1348, 2005.

[6] F. Denis, C. D'Halluin, and R. Gilleron. PAC learning with simple examples. In *STACS*, pages 231–242, 1996.

[7] J.A. Feldman. Some decidability results on grammatical inference and complexity. *Information and Control*, 20:244–262, 1972.

[8] J.A. Feldman. Real language learning. In *ICGI*, pages 114–125, 1998.

[9] E.M. Gold. Language identification in the limit. *Information and Control*, 10:447–474., 1967.

[10] 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.

[11] H. Hamburger and K. Wexler. A mathematical theory of learning transformational grammar. *Journal of Mathematical Psychology*, 12:137–177, 1975.

[12] J. A. Collins Hill. *A Computational Model of Language Acquisition in the Two-year-old.* Indiana University Linguistics Club, Indiana, 1983.

[13] M. Li and P.M.B. Vitanyi. Learning simple concepts under simple distributions. *SIAM Journal of Computing*, 20(5):911–935, 1991.

[14] 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.

[15] T. Oates, D. Desai, and V. Bhat. Learning k-reversible context-free grammars from positive structural examples. In *ICML*, pages 459–465, 2002.

[16] J. Oncina and P. García. Identifying regular languages in polynomial time. In H. Bunke, editor, *Advances in Structural and Syntactic Pattern Recognition*, pages 99–108. World Scientific Publishing, Singapore, 1992.

[17] Y. Sakakibara. Efficient learning of context-free grammars from positive structural examples. *Information Processing Letters*, 97:23–60, 1992.

[18] Y Sakakibara. Recent advances of grammatical inference. *Theoretical Computer Science*, 185:15–45, 1997.

[19] A. M. Schaerlaekens. *The two-word sentence in child language development.* Mouton, The Hague, 1973.

[20] I.M. Schlesinger. Production of utterances and language acquisition. In D.I. Slobin, editor, *The Ontogenesis of Grammar*, pages 63–103. Academic Press, New York-London, 1971.

[21] J.M. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61:39–61, 1996.

[22] 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.

[23] I. Tellier. Meaning helps learning syntax. In *ICGI*, pages 25–36, 1998.

[24] Cynthia A. Thompson and Raymond J. Mooney. Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, 18:1–44, 2003.

[25] T. Yokomori. Polynomial-time learning of very simple grammars from positive data. In *COLT '91: Proceedings of the fourth annual workshop on Computational learning theory*, pages 213–227. Morgan Kaufmann Publishers Inc., 1991.

A Natural Language Test Data

For each of the languages considered, a table gives the (1) translations of the initial set of fifteen utterances and their corresponding situations, (2) the word co-occurrence classes for this data and the set of non-background predicates that occur every situation in which that word class is used, and (3) the meanings chosen for each word that occurs in the sample in the construction of the meaning function γ. For those languages for which the initial sample does not result in convergence, additional tables give the results for (2) and (3) after convergence. These results are discussed in Section 5.

utterance	situation
almouthallath	bbt
almouthallath alazrak	bbtlbrt
almouthallath alahmar ela alshamal mina almourabbaa alazrak	brtlbbs
aldaerah fouk almouthallath alakhdar	bbcabgt
aldaerah alhamraa ela alyameen mina aldaerah alkhadraa	bgclbrc
almouthallath fouk almourabbaa alahmar	bgtabrs
almouthallath alakhdar	bgtabrs
aldaerah alzarkaa	bbcabgt
almouthallath alahmar ela alyameen mina almouthallath alazrak	bbtlbrt
aldaerah alhamraa	brc
aldaerah fouk almourabbaa	bbcabrs
aldaerah ela alshamal mina almourabbaa	bgclbgs
aldaerah alzarkaa fouk almourabbaa	bbcabgs
aldaerah ela alshamal mina almouthallath	bbclbgt
almouthallath ela alyameen mina aldaerah	bbclbgt

class	predicates
(almouthallath)	((tr 1))
(alazrak)	((bl 1) (tr 1) (le 2) (re 1))
(alahmar)	((re 1) (tr 1))
(ela mina)	((le 2))
(alshamal)	((le 2))
(almourabbaa)	((sq 1))
(aldaerah)	((ci 1))
(fouk)	((ab 2))
(alakhdar)	((ab 2) (gr 1) (tr 1))
(alhamraa)	((ci 1) (re 1))
(alyameen)	((le 2))
(alkhadraa)	((gr 1) (ci 1) (le 2) (re 1))
(alzarkaa)	((bl 1) (ci 1) (ab 2) (gr 1))

word	meaning
almouthallath	(tr 1)
alazrak	(bl 1)
alahmar	(re 1)
ela	()
alshamal	(le 2)
mina	()
almourabbaa	(sq 1)
aldaerah	(ci 1)
fouk	(ab 2)
alakhdar	(gr 1)
alhamraa	(ci 1)
alyameen	((le r) 2)
alkhadraa	(ci 1)
alzarkaa	(ci 1)

Table 3: Arabic: results for initial sample

class	predicates
(aldaerah)	((ci 1))
(alkhadraa)	((gr 1) (ci 1))
(almourabbaa)	((sq 1))
(ela mina)	((le 2))
(alyameen)	((le 2))
(alakhdar)	((gr 1))
(almouthallath)	((tr 1))
(alazrak)	((bl 1))
(fouk)	((ab 2))
(alshamal)	((le 2))
(alhamraa)	((ci 1) (re 1))
(alzarkaa)	((bl 1) (ci 1))
(alahmar)	((re 1))

word	meaning chosen
aldaerah	(ci 1)
alkhadraa	(gr 1)
almourabbaa	(sq 1)
ela	()
alyameen	((le r) 2)
mina	()
alakhdar	(gr 1)
almouthallath	(tr 1)
alazrak	(bl 1)
fouk	(ab 2)
alshamal	(le 2)
alhamraa	(ci 1)
alzarkaa	(ci 1)
alahmar	(re 1)

Table 4: Arabic: results after convergence; *alkhadraa*, *alhamraa* and *alzarkaa* not sufficiently resolved

utterance	situation
go to the triangle	bbt
go to the blue triangle	bbtlbrt
go to the blue square and then west to the red triangle	brtlbbs
go to the green triangle and then north to the circle	bbcabgt
go to the green circle and then east to the red circle	bgclbrc
go to the red square and then north to the triangle	bgtabrs
go to the the green triangle	bgtabrs
go to the blue circle	bbcabgt
go to the blue triangle and then east to the red triangle	bbtlbrt
go to the red circle	brc
go to the square and then north to the circle	bbcabrs
go to the square and then west to the circle	bgclbgs
go to the square and then north to the blue circle	bbcabgs
go to the triangle and then west to the circle	bbclbgt
go to the circle and then east to the triangle	bbclbgt

class	predicates
(go to the)	()
(triangle)	((tr 1))
(blue)	((bl 1))
(square)	((sq 1))
(and then)	()
(west)	((le 2))
(red)	((re 1))
(green)	((gr 1))
(north)	((ab 2))
(circle)	((ci 1))
(east)	((le 2))

word	meaning chosen
go	()
to	()
the	()
triangle	(tr 1)
blue	(bl 1)
square	(sq 1)
and	()
then	()
west	((le r) 2)
red	(re 1)
green	(gr 1)
north	((ab r) 2)
circle	(ci 1))
east	(le 2)

Table 5: Directions: results for initial sample have converged

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

class	predicates
(the)	()
(triangle)	((tr 1))
(blue)	((bl 1))
(red)	((re 1))
(to of)	((le 2))
(left)	((le 2))
(square)	((sq 1))
(circle)	((ci 1))
(above)	((ab 2))
(green)	((gr 1))
(right)	((le 2))

word	meaning chosen
the	()
triangle	(tr 1)
blue	(bl 1)
red	(re 1)
to	()
left	(le 2)
of	()
square	(sq 1)
circle	(ci 1)
above	(ab 2)
green	(gr 1)
right	((le r) 2)

Table 6: English: results for initial sample have converged

utterance	situation
to trigono	bbt
to mple trigono	bbtlbrt
to kokkino trigono sta aristera tou mple tetragonou	brtlbbs
o kyklos pano apo to prasino trigono	bbcabgt
o kokkinos kyklos sta deksia tou prasinou kyklou	bgclbrc
to trigono pano apo to kokkino tetragono	bgtabrs
to prasino trigono	bgtabrs
o mple kyklos	bbcabgt
to kokkino trigono sta deksia tou mple trigonou	bbtlbrt
o kokkinos kyklos	brc
o kyklos pano apo to tetragono	bbcabrs
o kyklos sta aristera tou tetragonou	bgclbgs
o mple kyklos pano apo to tetragono	bbcabgs
o kyklos sta aristera tou trigonou	bbclbgt
to trigono sta deksia tou kyklou	bbclbgt

class	predicates
(to)	()
(trigono)	((tr 1))
(mple)	((bl 1))
(kokkino)	((re 1) (tr 1))
(sta tou)	((le 2))
(aristera)	((le 2))
(tetragonou)	((le 2) (sq 1))
(o kyklos)	((ci 1))
(pano apo)	((ab 2))
(prasino)	((ab 2) (gr 1) (tr 1))
(kokkinos)	((ci 1) (re 1))
(deksia)	((le 2))
(prasinou)	((gr 1) (ci 1) (le 2) (re 1))
(kyklou)	((gr 1) (ci 1) (le 2))
(tetragono)	((ab 2) (sq 1))
(trigonou)	((bl 1) (tr 1) (le 2))

word	meaning chosen
to	()
trigono	(tr 1)
mple	(bl 1)
kokkino	(tr 1)
sta	()
aristera	(le 2)
tou	()
tetragonou	(sq 1)
o	(ci 1)
kyklos	()
pano	()
apo	(ab 2)
prasino	(gr 1)
kokkinos	(re 1)
deksia	((le r) 2)
prasinou	(re 1)
kyklou	(ci 1)
tetragono	(sq 1)
trigonou	(bl 1)

Table 7: Greek: results for initial sample

class	predicates
(to)	()
(mple)	((bl 1))
(trigono)	((tr 1))
(kokkino)	((re 1))
(tetragono)	((sq 1))
(sta tou)	((le 2))
(deksia)	((le 2))
(tetragonou)	((sq 1) (le 2))
(kyklou)	((ci 1) (le 2))
(pano apo)	((ab 2))
(o kyklos)	((ci 1))
(aristera)	((le 2))
(kokkinou)	((le 2) (re 1))
(trigonou)	((le 2) (tr 1))
(prasinou)	((le 2) (gr 1))
(kokkinos)	((re 1) (ci 1))
(prasino)	((gr 1))
(ton kyklo)	((ci 1) (ab 2))
(prasinos)	((gr 1) (ci 1))

word	meaning chosen
to	()
mple	(bl 1)
trigono	(tr 1)
kokkino	(re 1)
tetragono	(sq 1)
sta	()
deksia	((le r) 2)
tou	()
tetragonou	(sq 1)
kyklou	(ci 1)
pano	(ab 2)
apo	()
o	()
kyklos	(ci 1)
aristera	(le 2)
kokkinou	(re 1)
trigonou	(tr 1)
prasinou	(gr 1)
kokkinos	(ci 1)
prasino	(gr 1)
ton	()
kyklo	(ci 1)
prasinos	(ci 1)

Table 8: Greek: results after convergence; *kokkinos* and *prasinos* not sufficiently resolved

utterance	situation
hameshulash	bbt
hameshulash hacachol	bbtlbrt
hameshulash haadom mismol laribua hacachol	brtlbbs
haigul me'al lameshulash hayarok	bbcabgt
haigul haadom miyamin laigul hayarok	bgclbrc
hameshulash me'al laribua haadom	bgtabrs
hameshulash hayarok	bgtabrs
haigul hacachol	bbcabgt
hameshulash haadom miyamin lameshulash hacachol	bbtlbrt
haigul haadom	brc
haigul me'al laribua	bbcabrs
haigul mismol laribua	bgclbgs
haigul hacachol me'al laribua	bbcabgs
haigul mismol lameshulash	bbclbgt
hameshulash mismol laigul	bbclbgt

class	predicates
(hameshulash)	((tr 1))
(hacachol)	((bl 1))
(haadom)	((re 1))
(mismol)	((le 2))
(laribua)	((sq 1))
(haigul)	((ci 1))
(me'al)	((ab 2))
(lameshulash)	((bl 1) (tr 1))
(hayarok)	((gr 1))
(miyamin)	((le 2) (re 1))
(laigul)	((gr 1) (ci 1) (le 2))

word	meaning chosen
hameshulash	(tr 1)
hacachol	(bl 1)
haadom	(re 1)
mismol	(le 2)
laribua	(sq 1)
haigul	(ci 1)
me'al	(ab 2)
lameshulash	(tr 1)
hayarok	(gr 1)
miyamin	(re 1)
laigul	(gr 1)

Table 9: Hebrew: results for initial sample

class	predicates
(haribua)	((sq 1))
(hayarok)	((gr 1))
(miyamin)	((le 2))
(lameshulash)	((tr 1))
(haigul)	((ci 1))
(hacachol)	((bl 1))
(me'al)	((ab 2))
(laigul)	((ci 1))
(mismol)	((le 2))
(haadom)	((re 1))
(hameshulash)	((tr 1))
(laribua)	((sq 1))

word	meaning chosen
haribua	(sq 1)
hayarok	(gr 1)
miyamin	((le r) 2)
lameshulash	(tr 1)
haigul	(ci 1)
hacachol	(bl 1)
me'al	(ab 2)
laigul	(ci 1)
mismol	(le 2)
haadom	(re 1)
hameshulash	(tr 1)
laribua	(sq 1)

Table 10: Hebrew: results after convergence

utterance	situation
trikon	bbt
neela trikon	bbtlbrt
neele varg ke bain lal trikon	brtlbbs
hare trikon ke upar vritt	bbcabgt
hare vritt ke dain lal vritt	bgclbrc
lal varg ke upar trikon	bgtabrs
hara trikon	bgtabrs
neela vritt	bbcabgt
neele trikon ke dain lal trikon	bbtlbrt
lal vritt	brc
varg ke upar vritt	bbcabrs
varg ke bain vritt	bgclbgs
varg ke upar neela vritt	bbcabgs
trikon ke bain vritt	bbclbgt
vritt ke dain trikon	bbclbgt

class	predicates
(trikon)	((tr 1))
(neela)	((bl 1))
(neele)	((re 1) (tr 1) (le 2) (bl 1))
(varg)	((sq 1))
(ke)	()
(bain)	((le 2))
(lal)	((re 1))
(hare)	((ci 1) (gr 1))
(upar)	((ab 2))
(vritt)	((ci 1))
(dain)	((le 2))
(hara)	((gr 1) (tr 1) (ab 2) (re 1) (sq 1))

word	meaning chosen
trikon	(tr 1)
neela	(bl 1)
neele	(tr 1)
varg	(sq 1)
ke	()
bain	()
lal	(re 1)
hare	(gr 1)
upar	((ab r) 2)
vritt	(ci 1)
dain	(le 2)
hara	(sq 1)

Table 11: Hindi: results for initial sample

class	predicates
(trikon)	((tr 1))
(ke)	())
(dain)	((le 2))
(hara)	((gr 1))
(vritt)	((ci 1))
(varg)	((sq 1))
(upar)	((ab 2))
(neela)	((bl 1))
(hare)	((gr 1))
(lal)	((re 1))
(neele)	((bl 1))
(bain)	((le 2))

word	meaning chosen
trikon	(tr 1)
ke	()
dain	(le 2)
hara	(gr 1)
vritt	(ci 1)
varg	(sq 1)
upar	((ab r) 2)
neela	(bl 1)
hare	(gr 1)
lal	(re 1)
neele	(bl 1)
bain	((le r) 2)

Table 12: Hindi: results after convergence

utterance	situation
san jiao xing	bbt
lan se san jiao xing	bbtlbrt
lan se zheng fang xing zuo bian de hong se san jiao xing	brtlbbs
lü se san jiao xing shang mian de yuan xing	bbcabgt
lü se yuan you bian de hong se yuan	bgclbrc
hong se zheng fang xing shang mian de san jiao xing	bgtabrs
lü se san jiao xing	bgtabrs
lan se yuan	bbcabgt
lan se san jiao xing you bian de hong se san jiao xing	bbtlbrt
hong se yuan	brc
zheng fang xing shang mian de yuan	bbcabrs
zheng fang xing zuo bian de yuan	bgclbgs
zheng fang xing shang mian de lan se yuan	bbcabgs
san jiao xing zuo bian de yuan	bbclbgt
yuan you bian de san jiao xing	bbclbgt

class	predicates
(san jiao)	((tr 1))
(xing)	()
(lan)	((bl 1))
(se)	()
(zheng fang)	((sq 1))
(zuo)	((le 2))
(bian)	((le 2))
(de)	()
(hong)	((re 1))
(lü)	((gr 1))
(shang mian)	((ab 2))
(yuan)	((ci 1))
(you)	((le 2))

word	meaning chosen
san	()
jiao	(tr 1)
xing	()
lan	(bl 1)
se	()
zheng	(sq 1)
fang	()
zuo	((le r) 2)
bian	()
de	()
hong	(re 1)
lü	(gr 1)
shang	()
mian	((ab r) 2)
yuan	(ci 1)
you	(le 2)

Table 13: Mandarin: results for initial sample have converged

utterance	situation
triugolnik	bbt
siniy triugolnik	bbtlbrt
krasniy triugolnik c levo ot sinigo kvadrata	brtlbbs
krug nad zelonim triugolnikom	bbcabgt
krasniy krug c pravo ot zelonogo kruga	bgclbrc
triugolnik nad krasnim kvadratom	bgtabrs
zeloniy triugolnik	bgtabrs
siniy krug	bbcabgt
krasniy triugolnik c pravo ot sinigo triugolnika	bbtlbrt
krasniy krug	brc
krug nad kvadratom	bbcabrs
krug c levo ot kvadrata	bgclbgs
siniy krug nad kvadratom	bbcabgs
krug c levo ot triugolnika	bbclbgt
triugolnik c pravo ot kruga	bbclbgt

class	predicates
(triugolnik)	((tr 1))
(siniy)	((bl 1))
(krasniy)	((re 1))
(c ot)	((le 2))
(levo)	((le 2))
(sinigo)	((re 1) (tr 1) (le 2) (bl 1))
(kvadrata)	((le 2) (sq 1))
(krug)	((ci 1))
(nad)	((ab 2))
(zelonim triugolnikom)	((bl 1) (ci 1) (ab 2) (gr 1) (tr 1))
(pravo)	((le 2))
(zelonogo)	((gr 1) (ci 1) (le 2) (re 1))
(kruga)	((gr 1) (ci 1) (le 2))
(krasnim)	((gr 1) (tr 1) (ab 2) (re 1) (sq 1))
(kvadratom)	((ab 2) (sq 1))
(zeloniy)	((gr 1) (tr 1) (ab 2) (re 1) (sq 1))
(triugolnika)	((bl 1) (tr 1) (le 2))

word	meaning chosen
triugolnik	(tr 1)
siniy	(bl 1)
krasniy	(re 1)
c	()
levo	(le 2)
ot	()
sinigo	(re 1)
kvadrata	(sq 1)
krug	(ci 1)
nad	(ab 2)
zelonim	()
triugolnikom	(gr 1)
pravo	((le r) 2)
zelonogo	(gr 1)
kruga	(gr 1)
krasnim	(sq 1)
kvadratom	(sq 1)
zeloniy	(re 1)
triugolnika	(tr 1)

Table 14: Russian: results for initial sample

class	predicates
(krasniy)	((re 1))
(kvadrat)	((sq 1))
(krug)	((ci 1))
(siniy)	((bl 1))
(triugolnik)	((tr 1))
(c ot)	((le 2))
(pravo)	((le 2))
(triugolnika)	((tr 1) (le 2))
(nad)	((ab 2))
(krugom)	((ab 2) (ci 1))
(sinigo)	((bl 1) (le 2))
(kruga)	((ci 1) (le 2))
(zeloniy)	((gr 1))
(levo)	((le 2))
(zelonim)	((ab 2) (gr 1))
(kvadratom)	((sq 1) (ab 2))
(sinim)	((bl 1) (ab 2))
(triugolnikom)	((tr 1) (ab 2))
(krasnim)	((ab 2) (re 1))
(krasnogo)	((re 1) (le 2))
(kvadrata)	((sq 1) (le 2))
(zelonogo)	((gr 1) (le 2))

word	meaning chosen
krasniy	(re 1)
kvadrat	(sq 1)
krug	(ci 1)
siniy	(bl 1)
triugolnik	(tr 1)
c	()
pravo	((le r) 2)
ot	()
triugolnika	(tr 1)
nad	(ab 2)
krugom	(ci 1)
sinigo	(bl 1)
kruga	(ci 1)
zeloniy	(gr 1)
levo	(le 2)
zelonim	(gr 1)
kvadratom	(sq 1)
sinim	(bl 1)
triugolnikom	(tr 1)
krasnim	(re 1)
krasnogo	(re 1)
kvadrata	(sq 1)
zelonogo	(gr 1)

Table 15: Russian: results after convergence

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

class	predicates
(el)	()
(triangulo)	((tr 1))
(azul)	((bl 1))
(rojo)	((re 1))
(a la)	((le 2))
(izquierda)	((le 2))
(del)	()
(cuadrado)	((sq 1))
(circulo)	((ci 1))
(encima)	((ab 2))
(verde)	((gr 1))
(derecha)	((le 2))

word	meaning chosen
el	()
triangulo	(tr 1)
azul	(bl 1)
rojo	(re 1)
a	()
la	()
izquierda	(le 2)
del	()
cuadrado	(sq 1)
circulo	(ci 1)
encima	(ab 2)
verde	(gr 1)
derecha	((le r) 2)

Table 16: Spanish: results for initial sample have converged

utterance	situation
ucgen	bbt
mavi ucgen	bbtlbrt
mavi karenin solundaki kirmizi ucgen	brtlbbs
yesil ucgenin uzerindeki daire	bbcabgt
yesil dairenin sagindaki kirmizi daire	bgclbrc
kirmizi karenin uzerindeki ucgen	bgtabrs
yesil ucgen	bgtabrs
mavi daire	bbcabgt
mavi ucgenin sagindaki kirmizi ucgen	bbtlbrt
kirmizi daire	brc
karenin uzerindeki daire	bbcabrs
karenin solundaki daire	bgclbgs
karenin uzerindeki mavi daire	bbcabgs
ucgenin solundaki daire	bbclbgt
dairenin sagindaki ucgen	bbclbgt

class	predicates
(ucgen)	((tr 1))
(mavi)	((bl 1))
(karenin)	((sq 1))
(solundaki)	((le 2))
(kirmizi)	((re 1))
(yesil)	((gr 1))
(ucgenin)	((bl 1) (tr 1))
(uzerindeki)	((ab 2))
(daire)	((ci 1))
(dairenin)	((gr 1) (ci 1) (le 2))
(sagindaki)	((le 2))

word	meaning chosen
ucgen	(tr 1)
mavi	(bl 1)
karenin	(sq 1)
solundaki	((le r) 2)
kirmizi	(re 1)
yesil	(gr 1)
ucgenin	(bl 1)
uzerindeki	((ab r) 2)
daire	(ci 1)
dairenin	(gr 1)
sagindaki	(le 2)

Table 17: Turkish: results for initial sample

class	predicates
(mavi)	((bl 1))
(ucgenin)	((tr 1))
(sagindaki)	((le 2))
(ucgen)	((tr 1))
(daire)	((ci 1))
(yesil)	((gr 1))
(dairenin)	((ci 1))
(uzerindeki)	((ab 2))
(kirmizi)	((re 1))
(kare)	((sq 1))
(solundaki)	((le 2))
(karenin)	((sq 1))

word	meaning chosen
mavi	(bl 1)
ucgenin	(tr 1)
sagindaki	(le 2)
ucgen	(tr 1)
daire	(ci 1)
yesil	(gr 1)
dairenin	(ci 1)
uzerindeki	((ab r) 2)
kirmizi	(re 1)
kare	(sq 1)
solundaki	((le r) 2)
karenin	(sq 1)

Table 18: Turkish: results after convergence