A Language-agnostic Model of Child Language Acquisition

Louis Mahon[a], Omri Abend[b], Uri Berger[b,c], Katherine Demuth[d], Mark Johnson[e], Mark Steedman[a]

[a]*School of Informatics University of Edinburgh*
[b]*Computer Science and Engineering Hebrew University of Jerusalem*
[c]*School of Computing and Information Systems University of Melbourne*
[d]*Department of Linguistics Macquarie University*
[e]*School of Computing Macquarie University*

Abstract

This work reimplements a recent semantic bootstrapping child language acquisition (CLA) model, which was originally designed for English, and trains it to learn a new language: Hebrew. The model learns from pairs of utterances and logical forms as meaning representations, and acquires both syntax and word meanings simultaneously. The results show that the model mostly transfers to Hebrew, but that a number of factors, including the richer morphology in Hebrew, makes the learning slower and less robust. This suggests that a clear direction for future work is to enable the model to leverage the similarities between different word forms.

1. Introduction

This paper concerns computational models of CLA, which seek to understand the process of language acquisition by programming a computer to emulate the learning undergone by the child. When presented with data from a given language, such an algorithm should learn a degree of proficiency in that language. The fact that any child, when exposed to appropriate data, can learn any language establishes a strong connection between acquisition and language variation: whatever varies between languages must be

Email address: lmahon@ed.ac.uk (Louis Mahon)
URL: https://lou1sm.github.io/louismahonville/ (Louis Mahon)

specified by the data and must be learnable. It also makes it an essential requirement of a convincing CLA model that it be capable of learning any language. Here, we reimplement a recent computational CLA model (Abend et al., 2017), which is based on combinatory categorial grammar (Steedman, 2001) and semantic bootstrapping (Pinker, 1979), and is trained with an expectation-maximization style algorithm. This model is a suitable choice for understanding the acquisition process because of its cognitive plausibility. The dominant paradigm of large language models requires too much training data to be plausible models of how humans acquire language. Even on the small end of the scale they generally train on several orders of magnitude more tokens than a human sees in their entire life. Some have sought to better approximate human learning by learning from a more modest 10-100M tokens (Warstadt et al., 2023). However, such models still generally make a number of implausible design choices, such as multi-epoch training, batched parameter updates, and arbitrary text tokenization, and they do not, as we do, ensure that the training examples are presented in the order they appear to the child. Abend et al. (2017) in contrast is grounded in a well-developed theoretical model of semantic bootstrapping, and trains on each example only once, individually, in the order they appear to the child.

We test this model on two languages: English, on which it was originally tested, and Hebrew. The data we use is comprised of real child-directed utterances, taken from the CHILDES corpus (MacWhinney, 1998), coupled with a recent method for converting universal dependency annotations to logical forms (Szubert et al., 2024).

Firstly, we replicate the findings of Abend et al. (2017), and show that this model is successful in learning the important features of English syntax and semantics. We focus in the present paper on word order learning, and learning the meaning and syntactic categories of individual words. The results show that, after training, the model, correctly, strongly favours SVO order, predicts the right semantics for commonly appearing words and the right syntactic category for most. Then we apply the same training and testing procedure to the Hebrew corpus. There, the model learns word order and word meaning with a reasonably high accuracy. Its accuracy on syntactic categories is somewhat lower than that on English. We then discuss the difference in acquisition performance with respect to the linguistic differences between the two languages, and outline future extensions to the model that can more completely handle the learning of Hebrew without compromising the learning of English. Together these results demonstrate the model

in question is broadly successful in transferring between multiple languages, and support the argument that, in general for computational CLA models, it is important and instructive to evaluate on more than just a single language. The code for training and evaluation will be released on publication.

2. Method

2.1. Theoretical Underpinnings

Our model deals with syntax and semantics learning only. It assumes the child either has already learned to segment the speech stream and detect potential word boundaries, as evidenced in even young prelinguistic infants (Mattys et al., 1999), or is jointly learning phonotactics and morphology with syntax, as in the model of Goldberg and Elhadad (2013). At that point, the child must learn to combine atomic units (words) to produce a meaning representation that depends on (a) the meaning of the constituent words and (b) the manner in which they combine. Initially, for such a child, both are unknown. Theoretically, our approach to this problem falls under Semantic Bootstrapping Theory, (Pinker, 1979; Grimshaw, 1981; Brown, 1973; Bowerman, 1973; Schlesinger, 1971), which operates as follows: When a child hears an utterance "Bambi is home", we assume that, from a combination of perceptual context and background and innate knowledge, it can approximately identify the meaning of the entire utterance as some object in some state: $home(bambi)$. The task is then to figure out which words correspond to which parts of the meaning representation, and the language-specific principles by which they combine. As well as the correct interpretation, where English subjects precede VPs, others are also possible, e.g. "Bambi" means $home(\cdot)$, "is home" means *bambi* and subjects follow VPs. The original implementation of our model (Abend et al., 2017) designed a language learner that bootstraps learning of both (a) and (b) simultaneously.

2.2. Combinatory Categorial Grammar.

Combinatory categorial grammar (CCG) (Steedman, 2001) is a suitable theoretical framework for learning of this sort, because of its tight coupling of syntax and semantics. For each combination of syntactic categories, CCG provides a precise description of a corresponding semantic combination, $bambi + \lambda x.home(x) \rightarrow home(bambi)$. A full example is given for the two CHILDES (MacWhinney, 1998) corpora that we apply our model to: Adam (English), in Figure 1, and Hagar (Hebrew) in Figure 2.

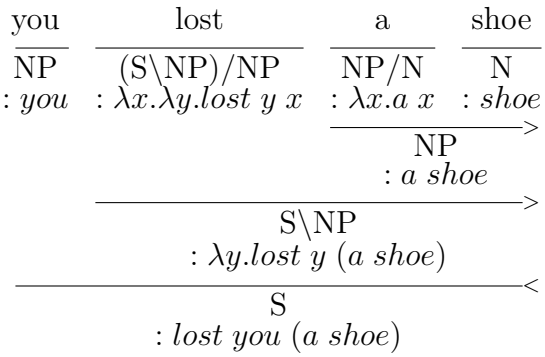


Figure 1: Example of a CCG derivation for a simple sentence from the Adam (English) corpus.

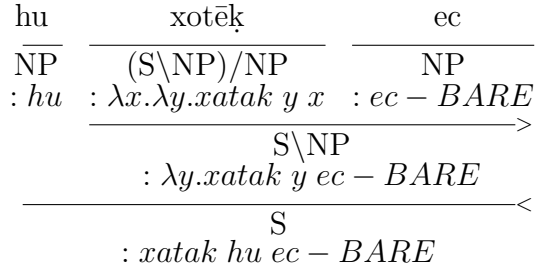


Figure 2: Example of a CCG derivation for a simple sentence from the Hagar (Hebrew) corpus. The sentence translates to English as "he's cutting wood", literally "he cut-pres-p wood".

Typically, CCG parsing is discussed in terms of combining constituents via combinatory rules to derive a root. For example, the last step of the derivation in Figure 1 uses the Backward Application Rule: $Y, X\backslash Y \rightarrow X$. Our learning model, when interpreting a sentence-meaning pair, runs these combinators in reverse, that is, it proceeds by successively splitting a root into smaller chunks until they can be aligned with word spans. We will thus often speak of CCG 'splits', by which we mean the CCG combinators run in reverse. The net effect is that our model considers all possible ways to split up the sentence and the meaning representation so that the semantic units best correspond to the language units.

2.3. Probabilistic Model

This section describes the details of our new implementation of the semantic bootstrapping CCG-based model of Abend et al. (2017).

For syntax and semantics learning, the three relevant aspects of each data point are the string of words in the utterance x, the meaning representation m, and the parse tree t. The former two are given by the data, but the parse tree is unobserved, and so treated as a latent variable. We assume that the data is drawn from some joint distribution $P(x, m, t)$, and we fit an approximation to this via several univariate conditional distributions. The conditional distributions are in the generative direction, i.e., together they produce a probability for the word string given the meaning representation. We make the Markovian assumption that the probability of splitting a node depends only on the syntactic category of that node, not on the rest of the tree or on the words or meaning. This means the parse tree can be modelled with a distribution of the form $p_t((s_1, s_2)|s)$, where s_1, s_2 and s are CCG syntactic categories. Note that the tuple (s_1, s_2) is ordered. This distribution also allows the possibility that s is a leaf node, the probability of which is expressed as $p_t(\text{leaf}|s)$.

We make similar Markovian assumptions on the relationship between syntactic category and meaning, and between meaning and word, so the distribution relating word x and meaning representation m has the form $p_w(x|m)$, and similarly for the distribution relating syntactic category s to meaning representation m. Following Abend et al. (2017), we add a second layer of prediction in the form of a shell logical form e, between the syntactic category and the logical form. The shell logical form replaces all non-variable terms with a placeholder marked for the function of the placeholder, e.g. verb, entity, determiner. The function is inferred from the CHILDES part-of-speech tag given in the method of Szubert et al. (2024). For example the logical form $\lambda y.lost\, y\, (a\, shoe)$ from Figure 1 has shell logical form $\lambda y.\text{vconst}\, y\, (\text{detconst nconst})$, because the CHILDES tags for *lost*, *a* and *shoe* are 'v', 'det:art' and 'n', respectively. See appendix for full list of conversions from CHILDES tags. This allows the model to share representation power for the structure of the logical form across different examples that may have different values for the constants. It is also used for the measure of word-order preference, described in Section 3.2. Thus, $p(m|s)$ is decomposed as $p_l(m|e)$ and $p_h(e|s)$. In $p_h(e|s)$, we ignore slash direction in s, so that e.g. conditioning on S\NP gives exactly the same distribution as conditioning on S/NP.

Each of these distributions is modelled as a Dirichlet process, to which Bayesian updates are applied at each data point. Taking p_w as an example,

the form of the posterior is then

$$p_w(x|m) = \frac{n(x,m) + \alpha H(x|m)}{n(m) + \alpha}\,, \tag{1}$$

where $n(x,m)$ is the number of times x and m have been observed together in the past, $n(m)$ is the number of times m has been observed in the past, and $H(x)$ is a pre-defined base distribution. An analogous definition holds for p_l, p_h and p_t. The alpha parameter is set to 1 for all distributions, corresponding to a uniform Dirichlet prior across simplices. During training, we set $\alpha = 10$ in p_t to encourage exploration of different syntactic structures, and $\alpha = 0.25$ in p_w to produce more confident predicted word meanings, which we find helps stabilize syntax learning.

2.4. Training Algorithm

The parameter updates described in Section 2.3 require tracking the occurrences on which two different elements co-occur. For example, in p_w, the probability of predicting the logical form $\lambda x.\lambda y.lost\,x\,y$ to be realized as the word 'lost' depends on the number of times that logical form and word were observed together during training. Because we do not observe parse trees directly, we instead employ an expectation-maximization algorithm, as follows. When the model observes a single data point X, consisting of an utterance as a string and a corresponding logical form, it uses its current parameter values $\theta^{(t)}$ to estimate a distribution over all possible parses that connect the two. The probability assigned to a parse tree T and the data point X is the following product

$$p(X,T|\theta^{(t)}) = \prod_{s'} p_t(s_1, s_2|s') \prod_{s} p_t(\text{leaf}|s) p_h(e_s|s) p_l(m_s|e_s) p_w(x_s|m_s)\,, \tag{2}$$

where s' ranges over all non-leaf nodes in T, s_1 and s_2 are the children of s', s ranges over all leaf nodes in T, and e_s, m_s and x_s are, respectively, the shell logical form, the logical form, and the word aligned to s in T. As we observe X, we are interested in the conditional probability of a given parse tree

$$p(T|X,\theta^{(t)}) = \frac{p(X,T|\theta^{(t)})}{\sum_{T' \in \mathcal{T}} p(X,T'|\theta^{(t)})}\,, \tag{3}$$

where $\mathcal{T}$ is the set of all allowable parses of X. For each parse tree, the co-occurrences that it gives rise to are recorded in proportion to the parse

tree's probability. Combining the standard expectation maximization (EM) update rule with the Bayesian update for the Dirichlet process, then, for each parameter θ that tracks the co-occurrence of two elements a and b, the update rule is given by

$$\theta^{(t+1)} = \theta^{(t)} + \mathbb{E}_{T \sim p(T|X,\theta^{(t)})}[\delta_T(a,b)]\,,$$

where $\delta_T(a,b)$ is an indicator function that is 1 if a and b co-occur in T and 0 otherwise.

The set $\mathcal{T}$ of allowable parse trees is the set of all valid CCG parse trees that have the observed logical form (LF) as root, the words in the observed utterance as leaves, and that have congruent syntactic and semantic types.[1]

This constraint is based on the assumption that the child knows the semantic type of a LF (or fragment thereof on some internal parse node). We approximate this knowledge using the part of speech (POS) tags as in the LF. The learner uses a mapping (shown in Appendix B) from these tags to semantic types. The tags were included in the original CHILDES transcription of the utterances, and maintained in the conversion procedure of Szubert et al. (2024), which generated our LFs. For example, the tag 'n:prop', indicating a proper noun, gets the semantic category e. In the case of ambiguous tags, such as the 'v' tag, which does not distinguish between transitive (<e,<e,t>>) and intransitive ($<e,t>$) verbs, we associate the node with a set of semantic types, and count it as permissible if its LF is congruent with any of these types. This use of CHILDES tags is as a proxy for semantic types of constants in the LF. The learner does not use them to infer part of speech. The full mapping from tags to types is given in Appendix B.

Type-raising (Steedman, 2001) introduces extra lambda variables into the LF, therefore, if a node has been type-raised, we count the slashes of the canonical non-type-raised version of its LF.

Adjectives are handled with the special 'hasproperty' predicate, which is given semantic category $<<<e,t>,<e,t>>,<e,t>>$, and requires exactly two lambda binders, and nouns are allowed to have no lambda binders and still be counted as permissable.

Additionally, we require the semantic category to be congruent with the CCG category, for each node. CCG's tight coupling of syntax and semantics provides a straightforward mapping from syntactic to semantic categories. In

[1] We use 'semantic/syntactic type' and 'semantic/syntactic category' interchangeably.

particular, the CCG atomic categories S and NP correspond to the Montagovian t and e respectively, and the slashes in non-atomic categories correspond to functions between types. For example, the CCG category $S\backslash NP/NP$ has the semantic type $<e,<e,t>>$. See Steedman (2001) for further details. If, when expanding a parse tree, any node violates these constraints, then that branch of search is terminated.

These constraints are an extension over the original model of Abend et al. (2017) They speed up training significantly. We do not have runtime figures for Abend et al., but based on our experiments, we find the speed-up to be at least 30x. They also make training more robust by removing the noise of updates from inconsistent parse trees. We believe this is the reason for our improved robustness to noise in the LFs, as described in Section 3.4.

The computation of all allowable parse trees can be performed efficiently by caching the probability of each subtree.

2.5. Worked Example

Here we present a worked example on a single training point. Recall that each training example consists of an utterance and corresponding logical form, and the learner considers the set $\mathcal{T}$ of all compatible parses, i.e. all parses with the observed LF as root, the observed utterance as leaves and that obeys the constraints described in Section 2.3. We describe the training updates for a single, correct parse for the example "you lost a pencil", as shown in Figure 3. For the purposes of this example, we show the exhaustive computation for every node in the tree. In practice the probabilities for upper nodes can be cached giving a large increase in efficiency.

The prediction of the tree proceeds from the root. First, the learner predicts a possible root category, here S with probability 0.388.

Next, it selects a possible split of the root syntactic category S, here the selected split is into NP and S\NP.

Then, for the left child node, it predicts the probability for a split into two daughter syntactic categories, or alternatively that this node is a leaf. Here, the prediction is that the node is a leaf, with probability 0.738. Then it predicts probabilities for a shell logical form given the syntactic category ('entity' with probability 0.66), a logical form given this shell LF (*you* with probability 0.327) and word (or span of words) given this LF ("you" with probability 0.912).

Meanwhile, for the right child of the root, it predicts, with probability 0.549, a split into further categories of S\NP/NP and NP.

Then for the left child of this node, it makes the equivalent predictions it made for the far left node, predicting a split or leaf given the syntactic category ('leaf' with probability 0.989), then of shell LF given syntactic category ($\lambda x.\lambda y.vconst\ x\ y$ with probability 0.961), of LF given shell LF ($\lambda x.\lambda y.lose_{past}\ x\ y$ with probability 0.012) and word span given LF ("lost" with probability 0.862).

For the right child of this node, the right-most NP in Figure 3, it predicts a split into NP/N and N, with probability 0.261.

For the NP/N daughter, it predicts 'leaf' with probability 0.994, then a shell LF given syntactic category ($\lambda x.quant\ x$ with probability 0.999), of LF given shell LF ($\lambda x.\lambda y.det : art|a; x$ with probability 0.486) and word span given LF ("a" with probability 0.95).

Finally for the N daughter node it predicts 'leaf' with probability 0.994, *noun* with probability 1.0 (note these figures are rounded to three places), $n|pencil$ with probability 0.015, and "pencil" with probability 0.973.

At this point, the semantic types of each node can be inferred. The rightmost $n|pencil$ node, in virtue of its n CHILDES pos tag, is inferred to have semantic type $<e,t>$. The $\lambda x.\ det{:}art|a\ x$ node to its left has type $<<e,t>,e>$. The parent of these two nodes then gets type $<e>$. Meanwhile the $\lambda x.\lambda y.v|lose_{past}\ x\ y$ has two allowable types based on the v, tag, $<e,t>$ and $<e,<e,t>>$. Only the latter is compatible with it having two lambda binders, so the former is discarded. Together with its sibling $<e>$ node, this gives its parent type $<e,t>$. Finally, together with its sibling, which is the leftmost leaf that has the tag 'pro:per' and so gets the semantic type $<e>$, these give the parent, which is the root node, type $<t>$. For all of these nodes, the semantic type is compatible with number of lambda binders and the CCG category, therefore the tree is used for training.

The total probability for this tree and leaves given the root LF is then computed by multiplying all the above probabilities:

$$0.388 \times 0.738 \times .66 \times .327 \times .912 \times .549 \times .961 \times .012 \times .862\times$$
$$\times .261 \times .994 \times .999 \times .486 \times .95 \times .994, \times 1.0 \times .015 \times .973 = 5.281e-7\,.$$

Given that we observe the leaves, we condition on this event by diving by the sum of the probabilities of all elements of $\mathcal{T}$. Here, that sum turns out to be $5.888e-7$, so the conditional probability for the tree in Figure 3 is

$$\frac{5.281e-7}{5.888e-7} \approx 0.896\,.$$

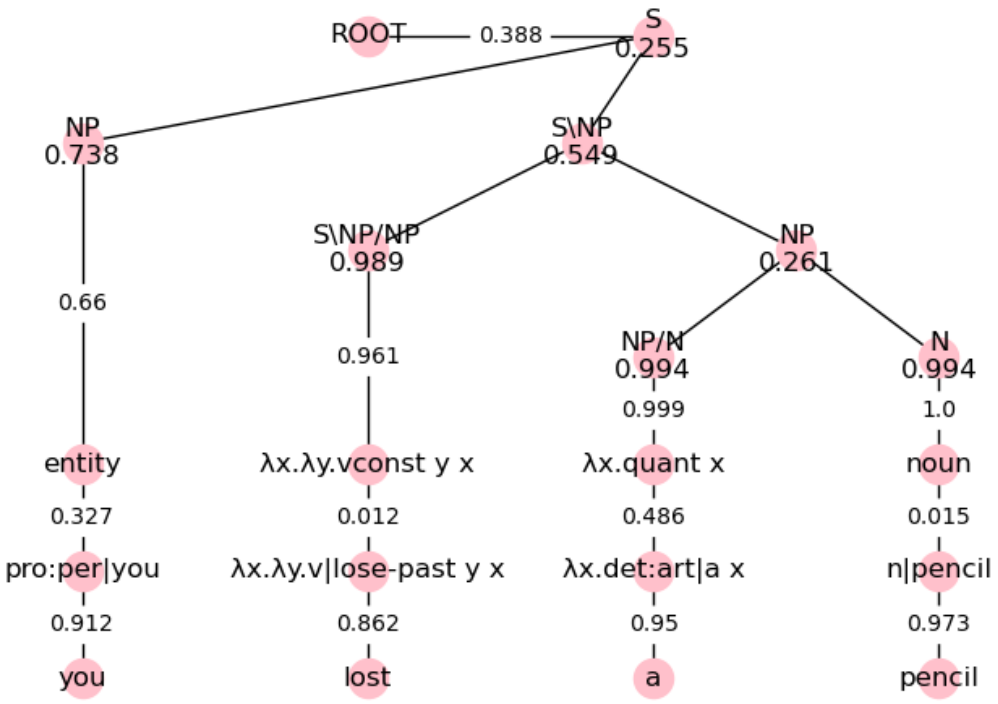


Figure 3: One of the parses considered by the learner for this example. Lambdas are written L and variables are numbers $0, \ldots,$.

Thus, in the Dirichlet processes that are used to make all model predictions, we update the counts of the co-occurrences in this tree by 0.896. An example of a co-occurrence in this tree is of the LF $you_{pro:per}$ with the word span "you". This is shown in the bottom left of Figure 3. The number of 'times' $you_{pro:per}$ has been observed to co-occurr with "you" is increased by 0.896.

This same procedure is repeated for every element of $\mathcal{T}$. In practice, for the corpora we use, this is generally $50 - 100$ trees.

3. Results

3.1. Datasets

In addition to the straightforward SVO examples in Figures 1 and 2, the LFs can express more complex syntactic features such as modals, negation,

utterance	**LF**
they 're in the drawer upstairs	$(upstairs\ (in\ (the\ drawer)\ they))$
penguins can't fly	$(\neg can\ (fly\ penguin_{pl})$
what are you giving them for dinner ?	$\lambda wh.for\ dinner\ (giving\ you\ wh\ them)$
get a kleenex and wipe your mouth	$get\ \_\ (a\ kleenex)\&wipe\ \_\ (your\ mouth)$

Table 1: Examples of more complex LFs that appear in our datasets.

prepositional phrases, and relative clauses. Table 1 shows some more complex examples. Further detailed examples can be found in Szubert et al. (2024), as well as full details of the process by which they were produced, and the rationale behind the various design choices involved in this production.

We post-process this data to exclude words that serve only as discourse markers and do not appear in the LFs or receive the CHILDES part-of-speech tag 'co', meaning 'communicator', which includes most instances of words like 'so', 'well' and the child's name. This results in 19314 tokens and 5320 utterances for Adam (English) and 6187 tokens and 3295 utterances for Hagar (Hebrew). As each data point contains exactly one utterance (as well as the corresponding LF), these are also the numbers of data points in each dataset.

3.2. Word Order

Following the procedure of Abend et al. (2017), we measure the acquisition of grammar and lexicon by examining the model's implicit word order parameters as a proxy. The degree to which the model favours each of the six possible word orders is determined by (a) the probability it assigns to the two CCG splits that are necessary to parse a simple transitive sentence under that order, and (b) the probability it assigns to the respective order in which the subject and object combine with the verb under that order. We assume that the verb-medial orders, SVO and OVS, must combine with the object first, so the six different possibilities are measured as follows:

$p(\text{SOV}) :=$
$p_t((\text{NP, S}\backslash\text{NP})|\text{S})p_t((\text{NP, S}\backslash\text{NP}\backslash\text{NP})|\text{S}\backslash\text{NP})p_h(\lambda x.\lambda y.\text{vconst } y\ x|(\text{S}\backslash\text{NP}\backslash\text{NP}))$

$p(\text{SVO}) :=$
$p_t((\text{NP, S}\backslash\text{NP})|\text{S})p_t((\text{S}\backslash\text{NP/NP, NP})|(\text{S}\backslash\text{NP}))p_h(\lambda x.\lambda y.\text{vconst } y\ x|(\text{S}\backslash\text{NP/NP}))$

$p(\text{VSO}) :=$
$p_t((\text{S/NP, NP})|\text{S})p_t((\text{S/NP/NP, NP})|\text{S/NP})p_h(\lambda x.\lambda y.\text{vconst } x\ y|(\text{S/NP/NP}))$

$p(\text{OSV}) :=$
$p_t((\text{NP, S}\backslash\text{NP})|\text{S})p_t((\text{NP, S}\backslash\text{NP}\backslash\text{NP})|\text{S}\backslash\text{NP})p_h(\lambda x.\lambda y.\text{vconst } x\ y|(\text{S}\backslash\text{NP}\backslash\text{NP}))$

$p(\text{OVS}) :=$
$p_t((\text{S/NP, NP})|\text{S})p_t((\text{NP, S/NP}\backslash\text{NP})|(\text{S/NP}))p_h(\lambda x.\lambda y.\text{vconst } y\ x|\text{S/NP }\backslash\text{NP})$

$p(\text{VOS}) :=$
$p_t((\text{S/NP, NP})|\text{S})p_t((\text{S/NP/NP, NP})|\text{S/NP})p_h(\lambda x.\lambda y.\text{vconst y x}|(\text{S/NP/NP}))\,.$

So, for example, the probability of SOV is product of the probability of splitting an S into NP and $S\backslash NP$, the probability of further splitting the resulting $S\backslash NP$ into NP and $S\backslash NP\backslash NP$, and the probability of this $S\backslash NP\backslash NP$ node having a logical form that takes two arguments and places the first in the object position and the second in subject position. (We use the convention that the argument immediately to the right of the verb is the subject.) The third term is what distinguishes $p(\text{SOV})$ from $p(\text{OSV})$.

Figures 4 and 5 take the relative values of these six scores, by normalizing so they sum to 1, and show how the above measure of word order preference changes over the course of training, for Adam (English) and Hagar (Hebrew) respectively. In both cases, the model succeeds in learning the correct SVO order, but this is faster and more extreme in Adam (English). In Hagar (Hebrew), the learning curve also appears somewhat step-shaped, with the SVO probability jumping up at a number of points, rather than increasing smoothly.

We hypothesize that, when learning an ordered category for an unknown word such as a transitive verb, the majority of early learning takes place in

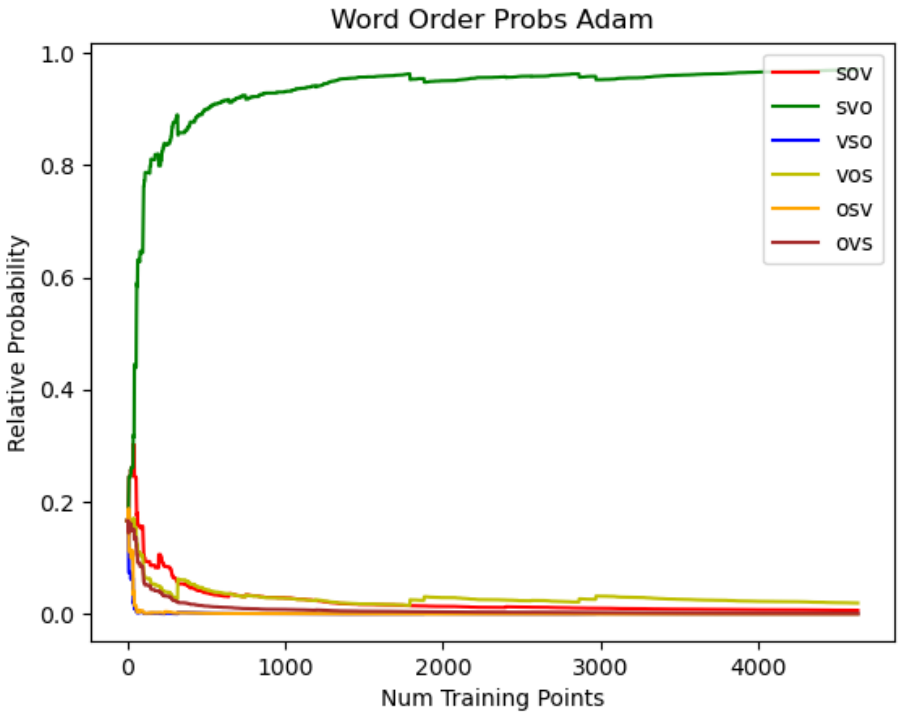


Figure 4: Evolution, over the course of training, of the learner's preference for each of the six possible word orders on Adam (English). It learns rapidly and confidently to favour SVO.

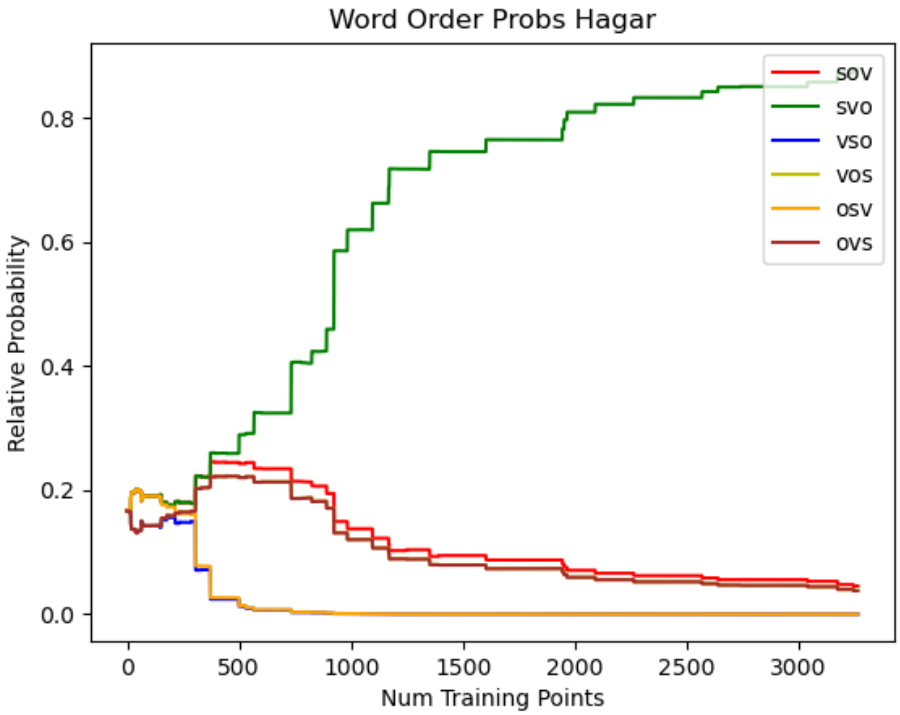


Figure 5: Evolution, over the course of training, of the learner's preference for each of the six possible word orders on Hagar (Hebrew). It learns SVO confidently, but more gradually than on Adam (English), and there are visible jumps where the learner encountered data points that were key for syntax learning.

dataset	num. critical examples	total num. word repeats	percentage new words	Zipf coefficient
Adam (English)	391	13744	7.99	1.436
Hagar (Hebrew)	14	3880	21.94	1.566

Table 2: Comparison of the diversity of words forms between the two datasets.

a small number of critical examples in which the phenomenon in question is clearly attested, and the child has already learned all the other words. In this case, we should therefore expect a noticeable rise in the probability of the correct order (SVO) on data points which (a) include a transitive meaning representation, (b) have no complicating features such as prepositional phrases, adverbials, reduplication or repetition, and (c) contain only words that the child has already encountered on previous data points. Tracking the number of such points, we find that there are 391 for Adam and 14 for Hagar. In Figure 7, we plot learning over just the first 365 points of Adam, as that also gives exactly 14 critical examples. Comparing Figure 5 (full Hagar dataset) and Figure 7 (first 365 data points of Adam), which both contain the same number of critical examples, we see that they both produce roughly the same shaped learning curves.

The apparent difference in the number of critical examples between Adam and Hagar is explained in part by the fact that the richer morphology leads to more diverse word forms and hence fewer examples when all words have been previously seen. This is measured explicitly, in various ways, in Table 2. As well as the count of critical examples, it shows (column 2) the total number of times any word repeats (column 2), the percentage of all words in all utterances that have not been seen before (column 3), and the Zipf coefficient (column 4).[2] For all four measures, we can see that Hagar (Hebrew) has more unseen words. Figure 6 evinces the same property graphically, by plotting the number of unique tokens (types) encountered as a function of the total

[2]The Zipfian distribution, which intuitively expresses that a small number of words account for the majority of word occurrences, with a long sparse tail of rare words, has been observed to well-model many corpora, including child-directed speech (CDS) (Lavi-Rotbain and Arnon, 2023). Formally, the occurrence frequency of the nth most common word is of the form $f_n = \frac{1}{(n+b)^a}$, where a and b are corpus-specific parameters, with a indicating the degree of sparsity and b determining the y-intercept. The Zipf coefficient we report is the a-parameter after fitting a function of this form to predict f_n from n.

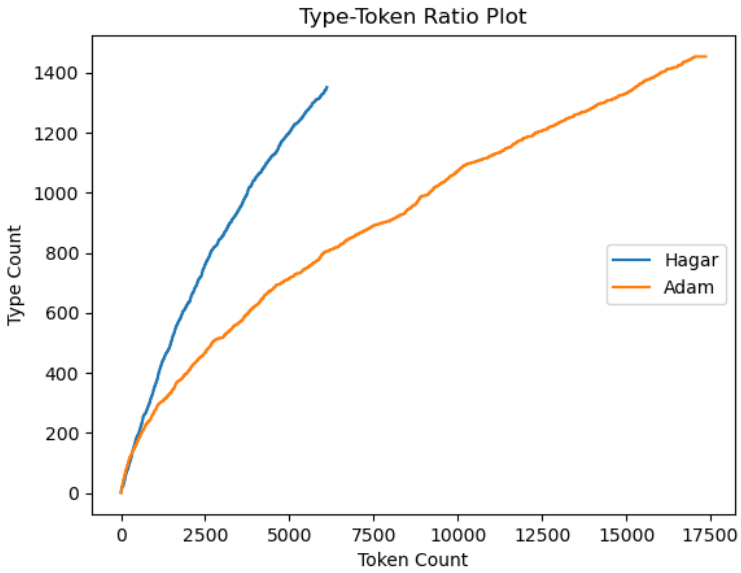


Figure 6: The number of unique types encountered throughout training. Hagar encounters more types for the same number of tokens, consistent with a greater diversity of word forms.

number of tokens encountered. That is, the plot passes through point (x, y) if and only if, after having seen exactly x tokens, the model has seen exactly y unique tokens. The steeper rise for Hagar (Hebrew) shows that it encounters more unseen word forms.

As well as the greater diversity of word forms, the less smooth learning curve for Hagar (Hebrew) in Figure 5 is also an artefact of the fact that we have chosen, following Abend et al. (2017), to report the measurement of SVO described in Section 3.2 as a proxy for degree of learning of the grammar as a whole. Although Hebrew is classified as an SVO language, like English, this proxy gives a misleading impression of the course of learning

Firstly, in English, adjective predication ("that's dangerous") and statements of membership ("he's a man") and identity ("that's Daddy") use a copula, providing evidence for SVO structure. The LFs for these sentences are 'hasproperty(that,dangerous)', 'equals(that, (a man))' and 'equals(that, Daddy)', respectively, so the model can easily learn that two meaning representations for 'is' are 'lambda x.lambda y.hasproperty y x' 'lambda x.lambda y.equals y x'. Hebrew, on the other hand, expresses these meanings without

copulae. For example: " at mecunēnet"–literally "you sick", or "ze cnonīt", meaning "this is a small radish"–literally "this small-radish". Copular sentences are highly frequent in both corpora.

Relatedly, many examples in the Hagar corpus are one- or two-word utterances, 73.9% vs 15.2% for Adam and, of course, sentences of fewer than 3 words cannot exhibit full SVO structure. The most common utterance in Hagar is "naḳōn", meaning "right/correct", which alone accounts for 10.2% of utterances. Finally, Adam is a larger dataset: 5320 vs 3295 data points.

We stress that these differences are not evidence for Hebrew being more difficult to learn in general than English. They mean only that the specific feature of SVO order that we are using as a proxy for overall learning is more strongly attested in English than Hebrew, as the two languages are represented in our datasets of Adam and Hagar. This is consistent with the idea Hebrew is, compared with languages such as English, less strongly committed to SVO structure Doron (2000).

3.3. Word Meaning and Syntactic Category

Going beyond this emergent favouring of SVO word order, what we are ultimately interested in learning is the lexicon, which relates words to pairs of syntactic categories and meaning representations. To evaluate this, we measure the model's prediction for logical form and syntactic category. For each dataset, we select the 50 most common words, and annotate them with a ground-truth logical form and syntactic category. The full CCG lexicon could contain many possibilities for each word, but we restrict to those that are attested in our corpora. See appendix for full list.

We then extract a predicted logical form m' for each word x as follows:

$$\begin{aligned} m' = \operatorname*{argmax}_m P(m|x) = \operatorname*{argmax}_m \frac{P(x|m)P(m)}{P(x)} = \operatorname*{argmax}_m P(x|m)P(m) \approx \\ \approx \operatorname*{argmax}_m p_w(x|m)p_w(m) = \operatorname*{argmax}_m p_w(x,m)\,, \end{aligned} \tag{4}$$

where the last quantity is determined by a Dirichlet process and so can be approximated by the observed number of times that w and m co-occur (recall this is in fact the sum of the expected values of their co-occurrence across all data points).

Similarly, we extract a predicted syntactic category for each word as

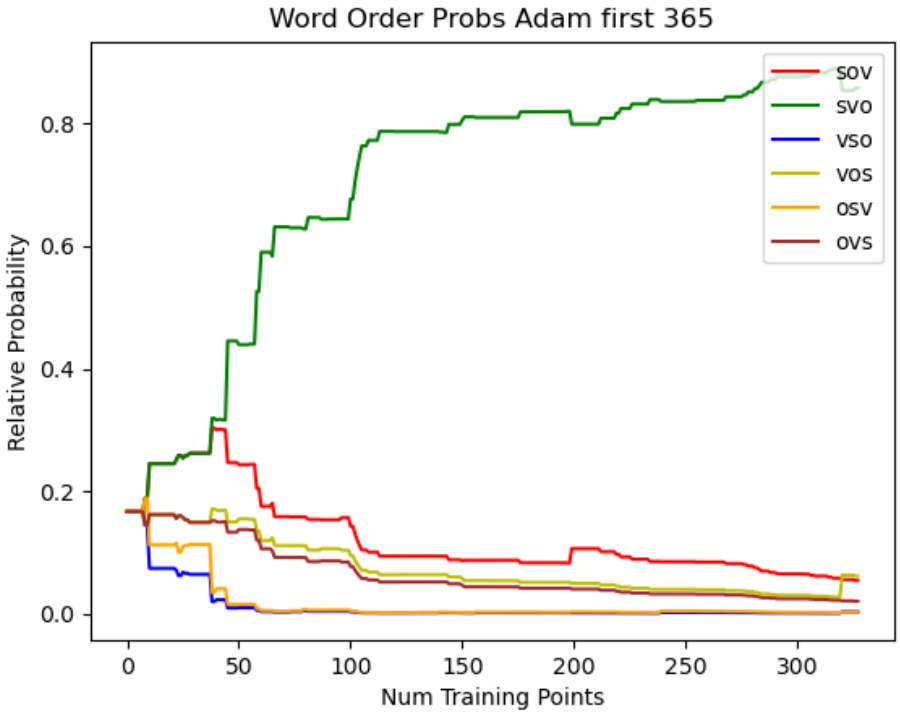


Figure 7: Zoomed in version of the first 365 data points in Adam (English), which contains the same number of critical examples as the full Hagar (Hebrew) dataset. The overall shape is similar to that of Hagar (Hebrew).

Corpus	meaning correct	syntactic category correct	both correct
Adam (English)	100%	76%	76%
Hagar (Hebrew)	100%	46%	34%

Table 3: Accuracy of the learned word meanings and syntactic categories on the fifty most common words, with respect to the manually annotated ground truth.

follows:

$$s' = \operatorname*{argmax}_s P(s|x) = \operatorname*{argmax}_s \frac{P(x|s)P(s)}{P(x)} = \operatorname*{argmax}_s P(x|s)P(s) = \\ = \operatorname*{argmax}_s \sum_m \sum_e P(x, m, e|s)P(s) \approx \sum_m \sum_e p_w(x|m)p_l(m|h)p_h(e|s)p_{syn}(s)\,, \tag{5}$$

and again, the last quantity can be computed straightforwardly, this time as a product of terms from each of the model's Dirichlet processes.

Table 3 reports the percentage of points for which the predicted meaning representation, as per (4) and the predicted syntactic category, as per (5) agree with the manually annotated ground truth. For both datasets, the model achieves 100% on the meaning representation, meaning it pairs every word with the correct logical form. The syntactic category accuracy is lower, 76% for Adam (English) and 46% for Hagar (Hebrew)[3]. This reflects the fact that the process of extracting the syntactic category is more involved than that for meaning. The reason the model can predict the correct meaning but the wrong syntactic category is that it first predicts a distribution over the former, and then uses this distribution to predict the latter, as in (5). If the model's prediction goes wrong at the second stage, then the meaning will be correctly predicted but the syntactic category will not.[4]

[3]Note that the accuracy for both can be lower than that for syntactic category, even with 100% meaning accuracy. This is because it is possible for the model to get both syntactic category and meaning right, but not get the pair right, for example, if it predicts 'NP: $\lambda x.run\ x$', then it is mixing up the syntactic category for run (noun) with the meaning representation for run (verb).

[4]In future work we plan to evaluate the course of learning more fully as the acquisition as a grammar and parsing model, incrementally training on weeks $1, \ldots, n$ and testing on week $n + 1$, using precision and recall of meaning representations as a measure, following Kwiatkowski et al. (2012).

3.4. Distractor Settings

To test the robustness of word-order learning to noise, we follow Abend et al. and consider a setting in which an utterance is paired not with a single logical form representing its meaning, but with multiple logical forms, only one of which represents its true meaning. The learner is then free to consider any of these LFs as the meaning of the utterance. Formally, what this means is that parse trees are computed for each of these LFs, and all of them are placed in the set $\mathcal{T}$. Then, when calculating the Bayesian posterior, as per (3), the denominator is larger, so the probability on any given tree is smaller, as compared to the no-distractor setting.

This makes learning more difficult, and simulates the fact that there may be some uncertainty for the child as to the meaning a given utterance represents. When there is a single tree that the model is very confident in, then the probability from this tree dominates anyway, and overall there is little effect from the distractor trees. However, when there is no such single confident interpretation, the distractor trees significantly reduce the probability on the trees from the correct LF, including the correct tree, and so dilute the learning effect.

The other, distractor, logical forms are taken from the utterances immediately following and preceding the given utterance. Specifically, the n distractor setting takes the $\lfloor n/2 \rfloor$ previous examples and the $\lceil n/2 \rceil$ following examples.

For example, in Adam, data points 226-228 are as follows:

Data point 226: "you blow it"–*blow you it*

Data point 227: "you can blow"–*can* (*blow you*)

Data point 228: "you do it"–*do you it*

Thus, in the two distractor setting, when training on data point 227, we include the parse trees from all three of these LFs. In this case, one possible interpretation takes the LF from data point 226–blow(you,it)–and interprets "you" as meaning $pro{:}per|you$, "can" as meaning $pro{:}per|it$, "blow" as meaning $\lambda x.\lambda y.v|blow\ y\ x$, and the sentence as being in SOV order. However, by this stage in training, the model has already learnt to place very little probability on the splits required for SOV, in particular splitting S\NP as NP + S\NP\NP, so this incorrect interpretation has a small probability and doesn't affect training much.

As shown in Figure 8, for Adam (English), the learner is still capable of learning the correct SVO order in all distractor settings. This is an improvement upon the version in Abend et al. (2017), where dealing with distractor

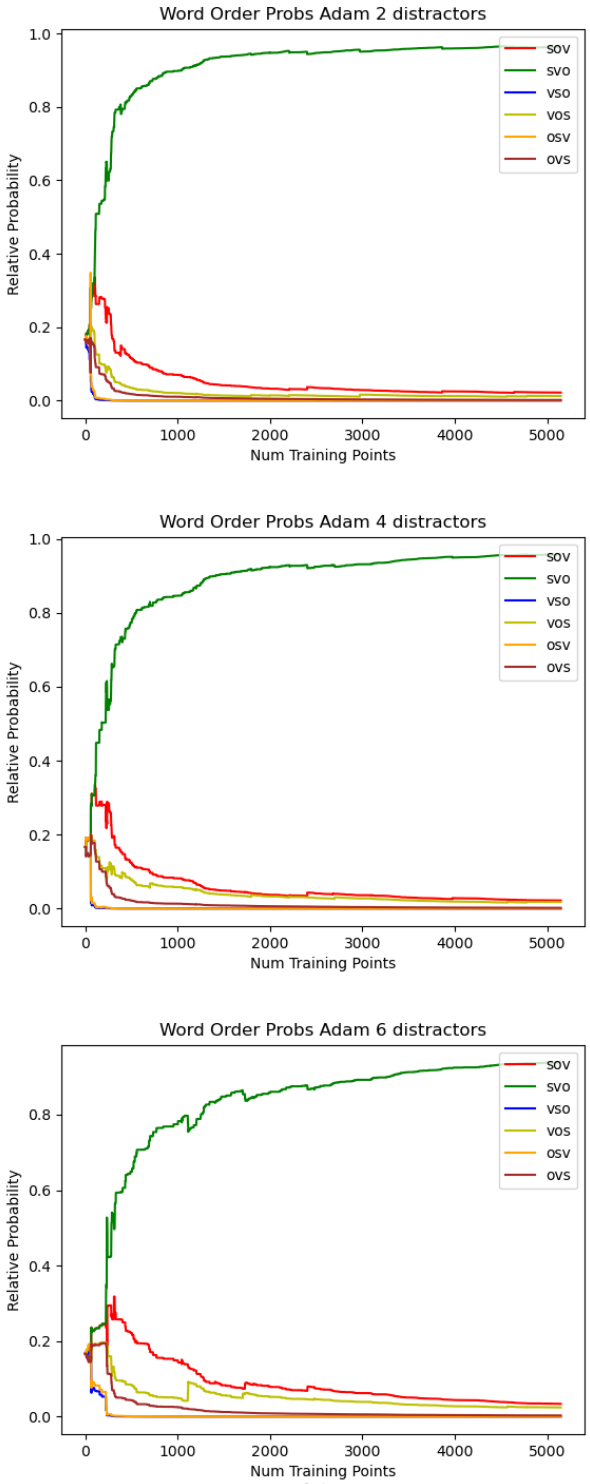


Figure 8: Word-order learning in Adam (English) with different numbers of distractor logical forms. More distractors slightly slows down learning, but the model still succeeds in confidently learning SVO.

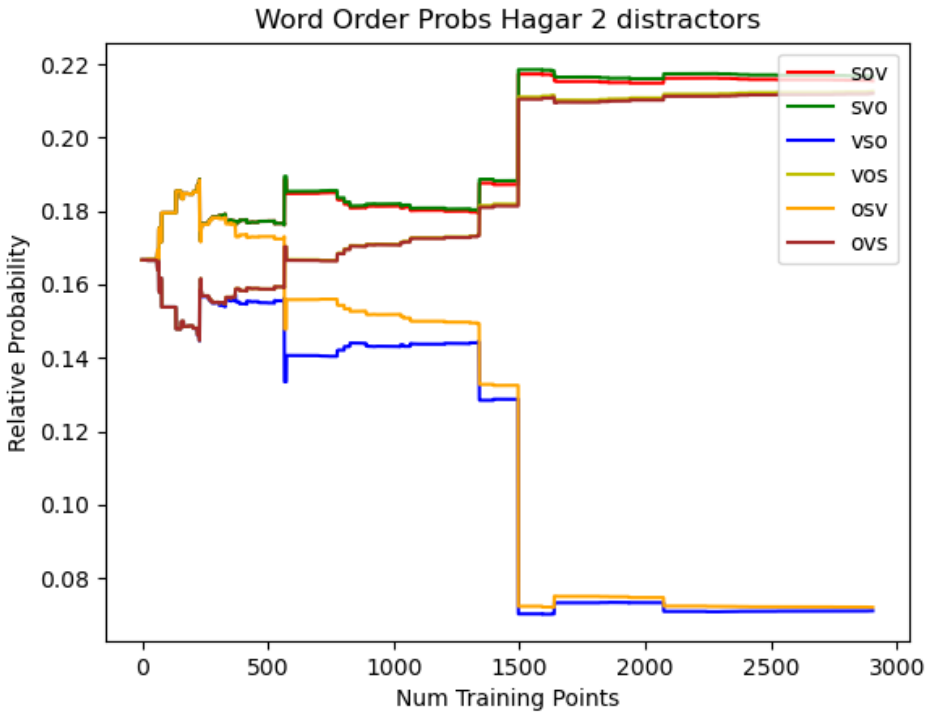


Figure 9: Word-order learning in Hagar (Hebrew) with two distractor logical forms. This prevents the learner from settling on SVO order, which does not rise to high probability and is only very marginally favoured over two of the other five orders.

settings required the introduction of an extra 'learning rate' parameter, that had to be set to different values for different numbers of distractors. In Appendix Appendix E, we present results for higher numbers of distractors, and show that, on Adam, the learner can handle up to 12 before performance starts to substantially degrade.

For Hagar (Hebrew), however, as shown in Figure 9, the model is not yet able to handle the distractor settings and fails to learn SVO. This fits with the picture, outlined above, that word-order is learnt correctly on Hagar (Hebrew), though currently not with as much confidence or robustness as on Adam (English).

Table 4 shows the performance in the distractor settings in terms of word meaning and syntactic category accuracy, as presented in Section 3.3. For both corpora, the accuracy drops as the number of distractors increases.

Corpus	meaning correct	syntactic category correct	both correct
Adam (English) 0 distractors	100%	76%	76%
Adam (English) 2 distractors	92%	62%	56%
Adam (English) 4 distractors	88%	42%	38%
Adam (English) 6 distractors	78%	32%	28%
Hagar (Hebrew) 0 distractors	100%	46%	34%
Hagar (Hebrew) 2 distractors	86%	36%	36%

Table 4: Extension of Table 3: word meaning and syntactic category accuracy in distractor settings.

Notably, the difference between the two corpora is much less striking than for word order learning: Adam (English) with six distractors is comparable to Hagar (Hebrew) with two distractors for word meaning and syntactic category learning, but, as seen by comparing Figures 8 and 9, it is much more successful at the presented measure of word order learning. This again suggests that the difference between Figures 8 and 9 is largely a result of SVO order (as it is measured here and in Abend et al. (2017)) being especially strongly attested in English, rather than the learner failing to acquire Hebrew in general.

4. Discussion

Our approach differs theoretically from other recent approaches to language acquisition. Ambridge (2020) argues that language acquisition can be understood purely based on the recall of all past occasions on which an utterance was used. This is claimed to adequately account for a range of linguistic phenomena without recourse to syntactic or semantic abstractions. Our model, on the other hand, uses abstractions in the form of lexical entries for words, and combinatory CCG rules. Chater and Christiansen (2018) treat language acquisition as the learning of a perceptuo-motor skill. One point of emphasis for Chater and Christiansen (2018) is the fact that much

relevant information to language learning is forgotten quickly, necessitating that learning occurs rapidly and in real-time (in this sense, the polar opposite of Ambridge (2020)). Another point of emphasis is the social context in which the child hears the utterance. We account for the first point by training on each example only once, one at a time, in the order they appear to the child. Pragmatic context is not currently represented in our input to the learner.

We also differ from these works in that ours is not purely theoretical but is based on a working model. An earlier work, Regier (2005) proposes a programmable model whose framework is similar to the theoretical account of Ambridge (2020). The data consists of utterances paired with manually created binary strings, where each bit indicates the presence or absence of a syntactic feature. Yang et al. (2002) proposes a probabilistic language acquisition model that assumes that the child begins with access to all possible grammars, which can be specified by a finite set of parameters, i.e. the principles and parameters framework (Chomsky, 1981; Hyams, 1986). It then learns to place more weight on those grammars that successfully parse observed sentences. This differs from our model, which acquires a statisical model of language-specific syntax, lexicon and logical form simultaneously by semantic bootstrapping from utterance meaning representations.

In the realm of word learning from speech, Räsänen and Khorrami (2019) presents a model for early word learning from real multimodal data, testing on English only.

Some neural bilingual CLA models have been proposed, which mostly focus on word-meaning learning, and do not attempt the more difficult task of learning from complete sentences. Li and Xu (2022) provide a summary of recent neural models for the related task of learning two languages simultaneously. The present study is, to our knowledge, the first cross-language evaluation of a computational semantic-bootstrapping model of how a child acquires language syntax and semantics.

As evidenced by our results, taking a model originally designed in the context of one language, and testing it on another, can reveal shortcomings which were obscured by the peculiarities of the first language. The main instance of this is the failure of the model to recognize similarities between word forms, which showed up in the learning of syntactic categories for individual words. There is a significantly lower accuracy in predicting Hebrew syntactic categories than English, though the predictions are still often correct. Further analysis reveals the lower accuracy to be largely due to Hebrew‘s richer

morphology producing a greater diversity of word forms. The model does not recognize these different forms as bearing any similarity, and instead, must learn each independently. This means it encounters fewer utterances where it already knows all word meanings and on the basis of which it can learn syntactic structure.

Testing on a second language also shows which aspects of the model are robust. In our case, the learning of word meanings transfers with high accuracy. Given that it was originally designed when only English data was available, and originally tested on only English data, this is a significant strength of the model. The learning of word order lies somewhere between the very successful learning of word meaning, and the less accurate learning of word syntactic category. The model does still confidently learn the correct SVO dominant order, but it is less robust to noise in the logical forms, and has a more jagged learning curve, and lower final confidence in SVO. This lower performance is noteworthy, but not insurmountable. Although Hebrew morphology is more complex than English, it has been shown to be learnable, by e.g. Goldberg and Elhadad (2013).

Hebrew is a suitable language to use as a first comparison to English, because the two have many, but not all, features in common.

5. Limitations

One limitation of our learner is that it does not model anything below the token level, the tokens being taken from the data of Szubert et al. (2024), which in turn took them from the CHILDES parses. The issue highlighted above of the sparsity of word forms suggests a future extension to allow it to guess a meaning for new words if they are similar in form to familiar words. This requires it to learn some internal structure to these tokens in virtue of which they can be similar or dissimilar to one another. One way to do this would be by explicitly adding morphology, and allowing the parse tree to extend not to word boundaries but to morpheme boundaries. Another option would be to add a neural word predictor. A character-level language model could be used to produce vectors for each word that depend on their structure, which could then be fed into a multi-layer perception. Different inflected forms of the same root should then have a similar word vector and so a similar predicted meaning. Note, this approach does not mean a replacement of anything that is currently in the model, it would model only morphology, not syntax or semantics. It may require more data but

potentially be more robust to the variety of inflected forms. Either of these additions could be learned independently of the model described here or in conjunction.

This need for the learner to discern a similarity between different inflected forms is obscured by the very sparse morphology in English. Different thematic roles for the same word or nominal phrase are not distinguished by morphological case as they are in Hebrew, so the model does not need to treat them as separate lexical entries. This further highlights the value of testing computational CLA models on multiple languages, and future work includes testing on further languages in addition to English and Hebrew.

Another limitation concerns our method of evaluating our model. Measuring the relative preference for different word orders allows comparison with Abend et al. (2017), but could give a misleading result in certain contexts where the correct analysis is a non-standard word order, e.g. in topicalization. In future, we hope to adopt a richer and more diverse evaluation suite, including measuring the fraction of test utterances with the correct inferred root LF and parse tree.

Thirdly, a more thorough evaluation of our learner involves testing on a more diverse set of languages, with larger and more comparable corpora. In contrast to the two corpora we use here, which differ in the number of tokens and utterances. It would also be interesting to compare different corpora for different children within the same language.

6. Conclusion

This paper reimplemented a recent computational model for child language acquisition, based on semantic bootstrapping, which learns from real transcribed child-directed utterances paired with annotated logical forms as meaning representations. We replicated the original results from this model on English, and performed the same evaluation on Hebrew. The results show that the ability of the model generally transfers well to a new language, but its learning on Hebrew is slower and less robust than on English. Further analysis reveals this, not surprisingly, to be due in large part to the richer morphology in Hebrew producing a more diverse set of word forms. Future work includes the extension of the model to detect and leverage similarities between word forms, application to other languages, and testing on corpora with equal numbers of utterances and tokens.

7. Acknowledgements

This research was supported by ERC Advanced Fellowship GA 742137 SEMANTAX and the University of Edinburgh Huawei Laboratory.

References

Abend, O., Kwiatkowski, T., Smith, N.J., Goldwater, S., Steedman, M., 2017. Bootstrapping language acquisition. Cognition 164, 116–143.

Ambridge, B., 2020. Against stored abstractions: A radical exemplar model of language acquisition. First Language 40, 509–559.

Bowerman, M., 1973. Structural relationships in children's utterances: Syntactic or semantic?, in: Moore, T. (Ed.), Cognitive Development and the Acquisition of Language. Academic Press, pp. 197–213.

Brown, R., 1973. A First Language: the Early Stages. Harvard University Press, Cambridge, MA.

Chater, N., Christiansen, M.H., 2018. Language acquisition as skill learning. Current opinion in behavioral sciences 21, 205–208.

Chomsky, N., 1981. Lectures on Government and Binding. Foris, Dordrecht.

Doron, E., 2000. Word order in hebrew, in: Lecarme, J., Lowenstamm, J., Shlonsky, U. (Eds.), Research in Afroasiatic Grammar. John Benjamins, pp. 41–56.

Goldberg, Y., Elhadad, M., 2013. Word segmentation, unknown-word resolution, and morphological agreement in a Hebrew parsing system. Computational Linguistics 39, 121–160.

Grimshaw, J., 1981. Form, function, and the language acquisition device. the logical problem of language acquisition, ed. by cl baker and john j. mccarthy, 165–182.

Hyams, N., 1986. Language Acquisition and the Theory of Parameters. Reidel, Dordrecht.

Kwiatkowski, T., Sharon, G., Zettlemoyer, L., Steedman, M., 2012. A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings, in: EACL.

Lavi-Rotbain, O., Arnon, I., 2023. Zipfian distributions in child-directed speech. Open Mind 7, 1–30.

Li, P., Xu, Q., 2022. Computational modeling of bilingual language learning: current models and future directions. Language Learning .

MacWhinney, B., 1998. The childes system. Handbook of child language acquisition , 457–494.

Mattys, S., Jusczyk, P., Luce, P., Morgan, J., 1999. Phonotactic and prosodic effects on word segmentation in infants. Cognitive Psychology 38, 465–494.

Pinker, S., 1979. Formal models of language learning. Cognition 7, 217–283.

Räsänen, O., Khorrami, K., 2019. A computational model of early language acquisition from audiovisual experiences of young infants. arXiv preprint arXiv:1906.09832 .

Regier, T., 2005. The emergence of words: Attentional learning in form and meaning. Cognitive science 29, 819–865.

Schlesinger, I., 1971. Production of utterances and language acquisition, in: Slobin, D. (Ed.), The Ontogenesis of Grammar. Academic Press, New York, pp. 63–101.

Steedman, M., 2001. The syntactic process. MIT press.

Szubert, I., Abend, O., Schneider, N., Gibbon, S., Mahon, L., Goldwater, S., Steedman, M., 2024. Cross-linguistically consistent semantic and syntactic annotation of child-directed speech. Language Resources and Evaluation , 1–50.

Warstadt, A., Mueller, A., Choshen, L., Wilcox, E., Zhuang, C., Ciro, J., Mosquera, R., Paranjabe, B., Williams, A., Linzen, T., Cotterell, R. (Eds.), 2023. Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, Association for Computational Linguistics, Singapore. URL: https://aclanthology.org/2023.conll-babylm.0.

Yang, C.D., et al., 2002. Knowledge and learning in natural language. Oxford University Press on Demand.

Appendix A. Conversion from CHILDES POS Tags to Shell LF Terms

As described in Section 2.3, we use the CHILDES part of speech tags, which are included in the logical forms of Szubert et al. (2024), to choose the marking on the constant in the shell logical form. Table A.5 gives full correspondence. In the main text in Section 3.2, we indicated the marking with the first letter of the right column, e.g. 'verb' gives 'vconst'.

Appendix B. Mapping from CHILDES POS Tags to Montagovian Semantic Types

Table B.6 shows how we infer the Montagovian semantic type from the CHILDES POS tags that are available in our LFs. Some are defined schematically, the avoid overly long expressions. For example, the category for conjunctions (conj) and coordinations (coord) are use the variable X to stand for any other semantic category. The reason the mapping from tags to semantic types is many-to-one is that this allows learning to be shared across categories. For example, if the model learns that the general category 'det' precedes nouns, it knows that this is true for all types of determiners, whereas if we distinguish between 'det:art', 'det:poss', 'det:num' etc., then it has to learn this separately for each.

Appendix C. Zipf Plots for Adam (English) and Hagar (Hebrew)

Section 3.2 reported the Zipf coefficient for the Adam (English) and Hagar (Hebrew) corpora. Here, Figure C.10 plots the word frequency against rank, both as observed in the data and as predicted by the fit Zipf function.

Table A.5: Our mapping from CHILDES part of speech tags of terms in the logical form to the marking on the constant in the corresponding shell logical form.

CHILDES TAG	const marking in shell LF
adj	adj
adv	adv
adv:int	adv
adv:tem	adv
aux	aux
conj	connect
coord	connect
cop	cop
det	quant
det:art	quant
det:dem	quant
det:int	quant
det:num	quant
det:poss	quant
mod	raise
mod:aux	quant
n	noun
n:pt	noun
n:gerund	entity
n:let	entity
n:prop	entity
neg	neg
prep	prep
pro:dem	entity
pro:indef	entity
pro:int	WH
pro:obj	entity
pro:per	entity
pro:poss	quant
pro:refl	entity
pro:sub	entity
qn	quant
v	verb

Table B.6: Our mapping from CHILDES part of speech tags of terms in the logical form to Montagovian semantic types.

CHILDES TAG	const marking in shell LF
adj	<<e,t>,<e,t>>
adv	not considered
adv:int	not considered
adv:tem	not considered
aux	not considered
conj	<X,<X,X>>
coord	<X,|<X,X>>
cop	handled separately
det	<<e,t>,e>
det:art	<<e,t>,e>
det:dem	<<e,t>,e>
det:int	<<e,t>,e>
det:num	<<e,t>,e>
det:poss	<<e,t>,e>
mod	<<<e,t>,<e,t>>,<e,t>>
mod:aux	<<e,t>,e>
n	<e,t>
n:pt	<e,t>
n:gerund	e
n:let	e
n:prop	e
neg	<<e,<e,t>>,<e,<e,t>>>, <<e,t>,<e,t>> t,t
prep	prep
pro:dem	e
pro:indef	e
pro:int	e
pro:obj	e
pro:per	e
pro:poss	<e,t>
pro:refl	e
pro:sub	e
qn	<e,t>
v	<e,<e,t>>, <e,t>

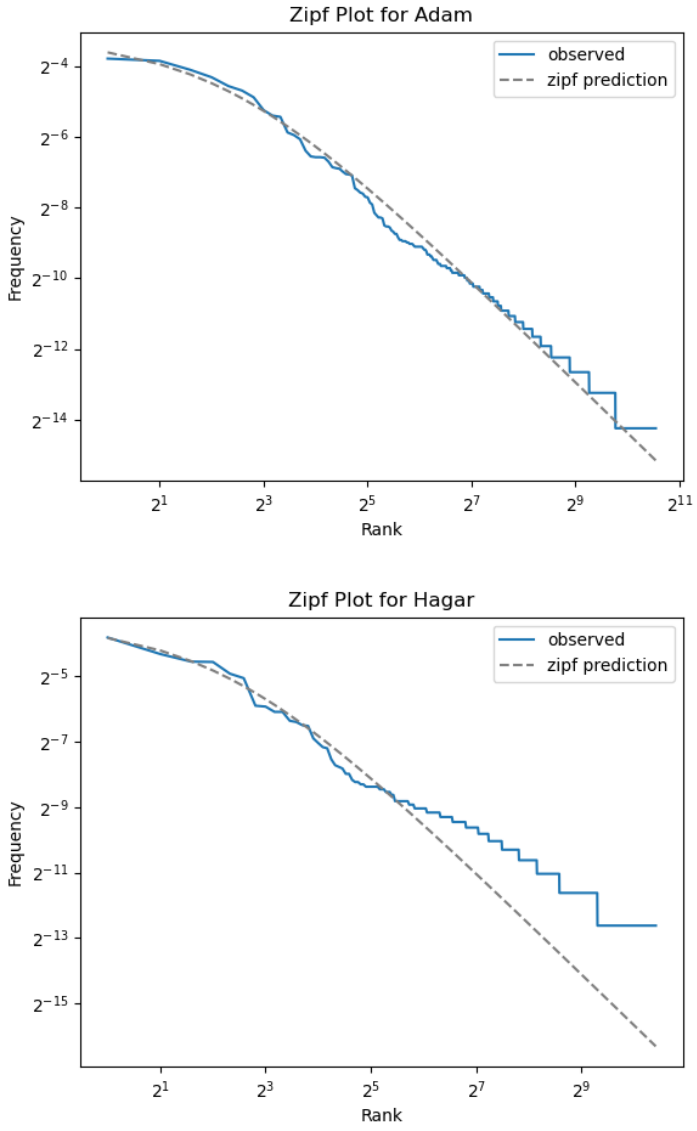


Figure C.10: Zipf plots for Adam (English) and Hagar (Hebrew), using the fit a and b parameters. Section 3.2 reports a as a measure of sparsity.

Appendix D. Logical Form and Semantic Category Accuracy by Word

Here we our ground truth annotation, and the learners' predictions, for each of the fifty most common words. The accuracy scores reported in Table 3 refer to the fraction of these words for which the prediction agrees with the the ground trtuh annotation.

Appendix D.1. Manually Annotated Lexicon for Fifty Most Common Words

This section shows the ground-truth logical form meaning representation and CCG syntactic category for the fifty most common words in each dataset. As described in Section 3.3, these are used to evaluate the learner's ability to acquire the correct lexicon. Note, the LFs that appeared in the main paper were abbreviated for clarity. Here, we write the full LF, including the CHILDES part of speech tag. The full lexical entry is of the form <LF> || <syntactic-category>. Where a word has two common meanings, we include two different lexical entries, separated with a comma.

Adam

```
'll: x. y.mod|~will (x y) || S\\NP/(S\\NP)
're: x. y.v|hasproperty y x || S\\NP/NP, x. y.v|equals y x || S\\NP/NP
's: x. y.v|equals y x || S\\NP/NP, x. y.v|hasproperty y x || S\\NP/NP
Adam:n:prop|adam || NP
I:pro:sub|i || NP
a: x.det:art|a x || NP/N
an: x.det:art|a x || NP/N
another: x.qn|another x || NP/N
are: x. y.v|equals x y || S\\NP/NP, x. y.v|hasproperty y x || S\\NP/NP
break: x. y.v|break y x || S\\NP/NP
can: x. y.mod|can (x y) || S\\NP/(S\\NP), x. y.mod|can (x y) || S/NP/(S\\NP)
d: x. y.mod|do (x y) || S\\NP/(S\\NP), x. y.mod|do (x y) || S/NP/(S\\NP)
did: x. y.mod|do-past (x y) || S/NP, x. y.mod|do-past (x y) || S/NP/(S\\NP)
do: x. y.v|do y x || S\\NP/NP, x. y.mod|do (x y) || S/NP/(S\\NP)
does: x. y.mod|do-3s (y x) || S\\NP/(S\\NP), x. y.mod|do-3s (x y) || S/NP/(S\\NP)
dropped: x. y.v|drop-past y x || S\\NP/NP
have: x. y.v|have y x || S\\NP/NP
he:pro:sub|he || NP
his: x.det:poss|his x || NP/N,pro:poss|his || NP
hurt: x. y.v|hurt-zero y x || S\\NP/NP
in: x. y.prep|in (y x) || S\\NP\\(S\\NP)/NP, x.prep|in x || S/S
is: x. y.v|equals x y || S\\NP/NP, x. y.v|hasproperty y x || S\\NP/NP
it:pro:per|it || NP
```

```
like: x. y.v|like y x || S\\NP/NP
lost: x. y.v|lose-past y x || S\\NP/NP
may: x. y.mod|may (x y) || S\\NP/(S\\NP)
missed: x.v|miss-past x || S\\NP, x. y.v|miss-past y x || S\\NP/NP
my: x.det:poss|my x || NP/N
name:n|name || N
need: x. y.v|need y x || S\\NP/NP
no: x.qn|no x || NP/N
not: x. y.not (x y) || S\\NP/(S\\NP)\(S\\NP/(S\\NP))
on: x.prep|on x || S\\NP\\(S\\NP)/NP
one:pro:indef|one || NP
pencil:n|pencil || N
say: x. y.v|say y x || S\\NP/NP
see: x. y.v|see y x || S\\NP/NP
shall: x. y.mod|shall (x y) || S\\NP/(S\\NP)
some: x.qn|some x || NP/N
that:pro:dem|that || NP, x.pro:det|that x || NP/N
the: x.det:art|the x || NP/N
they:pro:sub|they || NP
this:pro:dem|this || NP, x.pro:det|this x || NP/N
those:pro:dem|those || NP, x.pro:det|those x || NP/N
was: x. y.v|equals x y || S\\NP/NP, x. y.v|hasproperty y x || S\\NP/NP
we:pro:sub|we || NP
what:pro:int|WHAT || Swhq/Sq/NP,pro:int|WHAT || NP
who:pro:int|WHO || Swhq/Sq/NP,pro:int|WHO || NP
you:pro:per|you || NP
your: x.det:poss|your x || NP/N
```

Hagar

```
naḳōn:adv|naḳōn || S
ze: x.v|hasproperty pro:dem|ze x || NP
 at:pro:per| at || NP
ken:adv|ken || S
ha: x.det|ha x || NP/N
hu :pro:per|hu || NP
lo : x. y.not (x y) || S\\NP/(S\\NP)\(S\\NP/(S\\NP))
bō i:v|ba you || S
rocā: x. y.v|racā y x || S\\NP/NP
 anī:pro:per| anī || NP
 āba :n:prop| āba || NP
qxi:v|laqāx you || S
 od: x.qn| od x || NP/N
ṭa īm:adj|ṭa īm || S
ro ā: x.v|ra ā x || S\\NP
tistaklī:v|histakēl you || S
```

```
le:  x.prep|le x || S\\NP
hi :pro:per|hi  || NP
gamārnu:v|gamār you || S
hem:pro:per|hem || NP
nafāl:  x.v|nafāl x || S\\NP
texapṣī:v|xipēṣ you || S
kaxōl:adj|kaxōl || S
zo t:pro:dem|zo t || NP
lehitra ōt:v|hitra ā you || S
tir ī:v|ra ā you || S
beṭuxā:  x.v|hasproperty x adj|baṭūax || S\\NP
qar:adj|qar || S
 īma :n:prop| īma  || NP
 ōqer:adj| ōqer || S
halāḳ:  x.v|halāḳ x || S\\NP
glīda:n|glīda || N,n|glīda-BARE || NP
xam:adj|xam || S
 eyn:v| eyn you || S\\NP
boḳē:  x.v|baḳā x || S\\NP
yaldā:n|yēled || N,n|yēled-BARE || NP
gvinā:n|gvinā || N,n|gvinā-BARE || NP
tisperī:v|safār you || S
tarnegōl:n|tarnegōl || N,n|tarnegōl-BARE || NP
yeš:  x.v|yeš x || S\\NP
 avāl:  x.  y.v|hasproperty x y || S\\NP/NP
 or:n| or-BARE || NP
ricpā:n|ricpā || N,n|ricpā-BARE || NP
yēled:adj|yēled || N/N
 al:  x.prep| al x || S\\NP
 adōm:adj| adōm || S
tagīdi:v|higīd you || S
tašīri:v|šar you || S
cahōv:adj|cahōv || S
šalōm:n|šalōm || N
,n|šalōm-BARE || NP
```

Appendix D.2. Model Predictions

Tables D.7 and D.8 show the model predictions for Adam (English) and Hagar (Hebrew) respectively. In the interests of readability, we show only those words for which the predictions are not correct, either for the LF or for the syntactic category. For those words which are correctly predicted for both, the model predictions can be read off the ground-truth annotations, as provided in Section Appendix D.1.

	pred LF	pred syncat	LF correct	syncat correct
'll	$\lambda x.\lambda y.$mod| will $(x\ y)$	NP	True	False
can	$\lambda x.\lambda y.$Q (mod|can $(x\ y)$)	NP	True	False
d	$\lambda x.\lambda y.$Q (mod|do $(x\ y)$)	NP	True	False
did	$\lambda x.\lambda y.$Q (mod|do-past $(x\ y)$)	S\NP	True	False
does	$\lambda x.\lambda y.$Q (mod|do-3s $(y\ x)$)	NP	True	False
in	$\lambda x.$prep|in x	NP/N	True	False
like	$\lambda x.\lambda y.$v|like $y\ x$	NP	True	False
may	$\lambda x.\lambda y.$mod|may $(x\ y)$	NP	True	False
missed	$\lambda x.$v|miss-past x	NP	True	False
not	$\lambda x.\lambda y.$not $(x\ y)$	S\NP/NP	True	False
on	$\lambda x.$prep|on x	S\NP	True	False
shall	$\lambda x.\lambda y.$Q (mod|shall $(x\ y)$)	NP	True	False

Table D.7: List of all incorrect model LF and syntactic category predictions for English.

Appendix E. Plots of Higher Numbers of Distractors

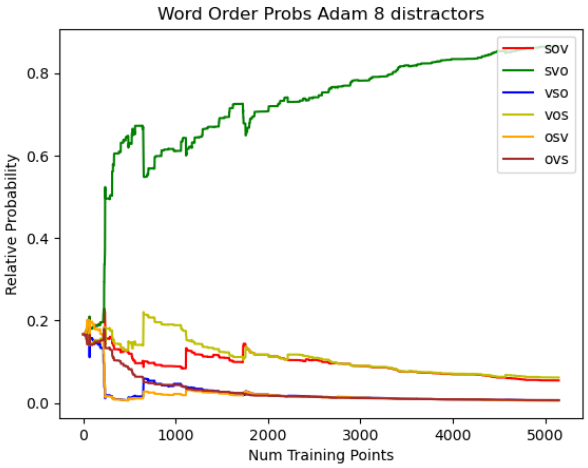
In Section 3.4, we following Abend et al. (2017) in reporting the trend of word-order learning in settings with 2, 4 and 6 distractor settings for English (for us, this is the Adam corpus, for Abend et al., this was the Eve corpus. For Hagar (Hebrew), we reported just 2 distractors, because already this was too much for the model to learn word order effectively, for the reasons discussed in Section 3.4. Here, we report high number of distractors for Adam: 8, 10 and 12, which shows further robustness to noise in the LFs when learning English word order. For 8 and 10 distractors, SVO is still learnt confidently and relatively smoothly. For 12 distractors, it takes much longer before SVO starts to dominate, but by the end, the learner has still quite firmly acquired SVO. Note that the differences in relative probability between the six orders are more significant towards the end of training, because by that time that model has seen more data and so it would take more data again for it to change its mind. Formally, the denominators in the Dirichlet processes have become large. Therefore, the spike of SVO at the end is more significant than the spike of OSV at the beginning.

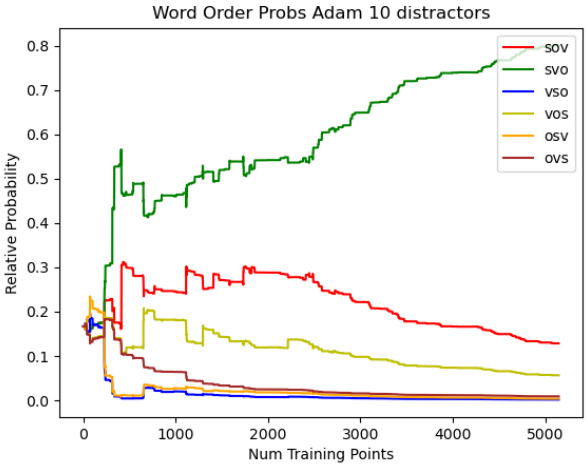
Appendix F. Plots vs Number of Tokens

Because the average number of tokens differs between the two corpora, one may also want to consider how learning develops as a function of the number of tokens seen rather than the number of utterances. This is shown in Figure F.12.

	pred LF	pred syncat	LF correct	syncat correct
naḳōn	adv|naḳōn	NP/N	True	False
ken	adv|ken	NP/N	True	False
lo	$\lambda x.\lambda y.$not $(x\ y)$	NP	True	False
bō i	v|ba you	NP	True	False
rocā	$\lambda x.\lambda y.$Q (v|racā $y\ x$)	Sq\NP	True	False
qxi	v|laqāx you	NP	True	False
ṭa īm	adj|ṭa īm	NP	True	False
ro ā	$\lambda x.$Q (v|ra ā x)	Sq\NP	True	False
tistaklī	v|histakēl you	NP	True	False
gamārnu	Q (v|gamār you)	NP	True	False
nafāl	$\lambda x.$Q (v|nafāl x)	Sq\NP	True	False
texapṣī	v|xipēṣ you	NP	True	False
kaxōl	adj|kaxōl	NP	True	False
lehitra ōt	v|hitra ā you	NP	True	False
tir ī	v|ra ā you	NP	True	False
beṭuxā	$\lambda x.$Q (v|hasproperty x adj|baṭūax)	Sq\NP	True	False
qar	adj|qar	NP	True	False
ōqer	adj| ōqer	NP	True	False
glīda	n|glīda	NP	True	True
xam	adj|xam	S/S	True	False
boḳē	$\lambda x.$Q (v|baḳā x)	Sq\NP	True	False
yaldā	n|yēled	NP	True	True
gvinā	Q (n|gvinā)	NP	True	True
tisperī	v|safār you	NP	True	False
tarnegōl	n|tarnegōl	NP	True	True
avāl	$\lambda x.\lambda y.$v|hasproperty $x\ y$	NP	True	False
ricpā	n|ricpā	NP	True	True
yēled	adj|yēled	NP	True	False
adōm	adj| adōm	NP	True	False
tagīdi	v|higīd you	NP	True	False
tašīri	v|šar you	NP	True	False
cahōv	adj|cahōv	NP	True	False
šalōm	n|šalōm	NP	True	True

Table D.8: List of all incorrect model LF and syntactic category predictions for English. Again note that, as we evaluate here, the model can get both LF and syntactic category correct individually but still get the overall prediction wrong if the two do not match.





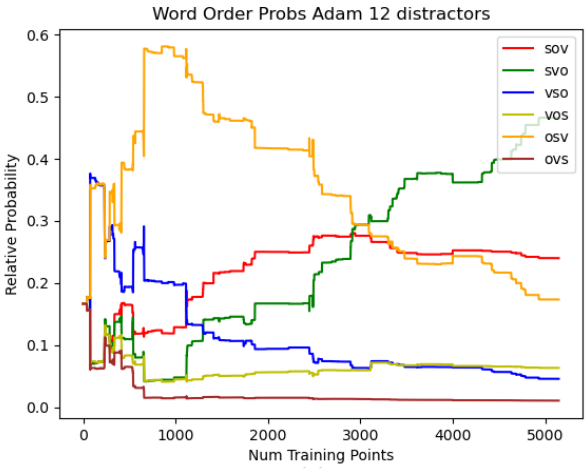


Figure E.11: Word-order learning in Adam (English) with higher numbers of distractor logical forms.

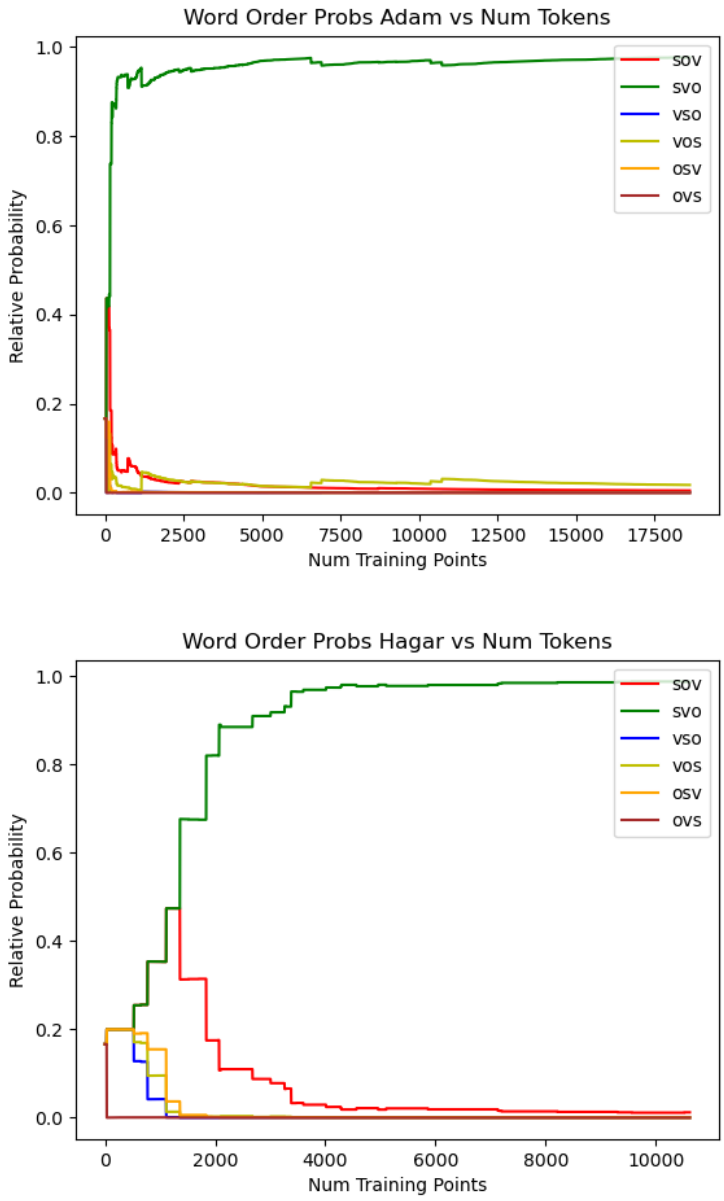


Figure F.12: Plot of the relative probability of the six word orders as a function of the number of tokens the model as seen during training.