Finding Structure in One Child’s Linguistic Experience

Wentao Wang, Wai Keen Vong, Najoung Kim, and Brenden M. Lake
Center for Data Science, New York University

Abstract

Neural network models have recently made striking progress in natural language processing, but they are typically trained on orders of magnitude more language input than children receive. What can these neural networks, which are primarily distributional learners, learn from a naturalistic subset of a single child’s experience? We examine this question using a recent longitudinal dataset collected from a single child, consisting of egocentric visual data paired with text transcripts. We train both language-only and vision-and-language neural networks and analyze the linguistic knowledge they acquire. In parallel with findings from Elman’s (1990) seminal work, the neural networks form emergent clusters of words corresponding to syntactic (nouns, transitive and intransitive verbs) and semantic categories (e.g., animals and clothing), based solely on one child’s linguistic input. The networks also acquire sensitivity to acceptability contrasts from linguistic phenomena such as determiner-noun agreement and argument structure. We find that incorporating visual information produces an incremental gain in predicting words in context, especially for syntactic categories that are comparatively more easily grounded such as nouns and verbs, but the underlying linguistic representations are not fundamentally altered. Our findings demonstrate which kinds of linguistic knowledge are learnable from a snapshot of a single child’s real developmental experience, and which kinds may benefit from stronger inductive biases or richer sources of data.

1 Introduction

In the first three years of life, children’s linguistic development progresses rapidly. Young children begin understanding words at around 6 months (Tincoff and Jusczyk, 1999, 2012; Bergelson and Swingley, 2012, 2015). The vocabulary that they can comprehend and produce increases gradually until around 12–14 months, at which a non-linear comprehension boost occurs (Bergelson, 2020) and lexical-semantic networks begin to develop (Wojcik, 2018). Language learning remains both a scientific and engineering puzzle; it is unclear what inductive biases are necessary and how much can be learned through relatively generic learning mechanisms, such as distributional learning from patterns of word co-occurrence (Firth, 1957; Harris, 1954; Landauer and Dumais, 1997).

To provide some insight into this learning challenge, we focus on accurately capturing a subset of the linguistic and visual inputs received by a single child during their development. We then train generic computational models without language-specific inductive biases on this data and evaluate what these models learn (e.g., Orhan et al. 2020). Previously, a major obstacle to this approach was the lack of high-quality and substantive developmental data. However, thanks to large-scale developmental datasets containing linguistic input (MacWhinney, 2000; Roy et al., 2015; Sullivan et al., 2021) and recent advances in deep learning, it is now possible to run large-scale simulations on real language input. Training neural networks on these datasets, and then analyzing what kinds of knowledge are acquired, can help to answer foundational questions about what is learnable from a child’s experience (Huebner and Willits, 2018; Warstadt and Bowman, 2022).

In this work, we follow this approach by using SAYCam, a recent longitudinal developmental dataset consisting of an egocentric visual and linguistic input to a single child spanning 6 to 25 months of age.
months of age (Sullivan et al., 2021). The scale of this dataset allows us to train several widely used neural network architectures and explore what they learn, in terms of how they structure their representations and how this affects behavior. The networks we adopt are not designed for human languages specifically; rather, they are configured to process general sequences. We first train two kinds of neural networks, Long Short-Term Memory (LSTM; Hochreiter and Schmidhuber, 1997) and Continuous Bag-Of-Words (CBOW; Mikolov et al., 2013), on only the language portion of the dataset and analyze the syntactic and semantic structure they acquire. Then, we add the visual data and train an image captioning model (Xu et al., 2015) on the paired vision-and-language dataset, and examine the impact on linguistic knowledge from incorporating the visual modality.

Our work builds on previous examinations into what neural networks can learn from linguistic input (Elman, 1990; Huebner and Willits, 2018; Huebner et al., 2021, i.a.). In his pioneering article, Elman (1990) formulated a means of training Simple Recurrent Networks (SRNs) to predict the next word in a sentence given the previous words. When applied to simple language-like inputs, these networks formed coherent clusters of words, analogous to real English syntactic and semantic categories. More recently, researchers have examined similar questions using naturalistic sources of data combined with more capable neural network architectures, such as LSTM (Hochreiter and Schmidhuber, 1997) and Transformer (Vaswani et al., 2017). For instance, Huebner and Willits (2018) trained both Elman’s SRNs and LSTMs on a corpus of naturalistic, developmental linguistic data (CHILDES; MacWhinney, 2000), and analyzed emergent clusters in their acquired representations. Similarly, Huebner et al. (2021) trained a Transformer on a corpus derived from CHILDES (AO-CHILDES; Huebner and Willits, 2021) and analyzed its syntactic knowledge. Our work follows this tradition that began with Elman. The most distinctive aspect of our work is that the networks are trained on a strict subset of real developmental experience of an individual child. Previous work in this vein aggregated linguistic input to multiple children; although this provides a larger corpus than SAYCam, it does not constitute what a single child would experience and thus does not directly address questions regarding learnability.

From our simulations and analyses, we have both positive and negative findings regarding learnability. When using language-only data, we find that networks can differentiate words in different syntactic categories, such as nouns, transitive and intransitive verbs, and semantic categories, such as animals and clothing. We also find that these networks acquire nascent syntactic abilities, such as inferring the syntactic category of a word from its context. In some cases, they can recognize determiner-noun agreement and argument structure regarding verb transitivity, but they struggle with other phenomena such as subject-verb agreement. Additionally, we find that introducing visual information provides an incremental improvement on our networks’ abilities to predict words in context, but does not fundamentally alter the linguistic representations.

2 Sensory Input Through the Eyes and Ears of a Child

In this section, we briefly describe the data streams used for training and evaluating our neural networks. The data is a subset of SAYCam (Sullivan et al., 2021), a dataset consisting of egocentric head-mounted camera recordings of 3 very young children. Each child’s recordings are longitudinal and recorded at regular intervals (several hours each week) for around 2 years starting from 6–8 months of age. However, out of the 3 children, only one (labeled as baby S) had a large proportion of his naturalistic speech input transcribed (spanning 6–25 months of age), making baby S the choice for our focus. This dataset, which we call the SAYCam-S dataset, consists of child-directed utterances paired with visual data from the child’s point of view at the time of the utterance.

We outline the major steps taken to preprocess the dataset. For each original transcript, we first replace anything annotated as “inaudible” with a special <UNK> (unknown) token, and use the spaCy tokenizer (Honnibal and Montani, 2017) to segment the inputs into discrete tokens. Moreover, long utterances were split into multiple sentences, and their time spans

---

1 As in previous work, we draw parallels between emergent clusters of word embeddings and real-world categories (“animal”, “vehicle”, etc.). Importantly, however, these learned representations are quite limited in function and structure compared to full-fledged human conceptual representations (Lake and Murphy, 2021). We elaborate on this point in the General Discussion.
<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of utterances</td>
<td>33737</td>
<td>1874</td>
<td>1875</td>
</tr>
<tr>
<td>Mean (SD) utterance length</td>
<td>6.67 (5.49)</td>
<td>6.59 (5.46)</td>
<td>6.62 (4.95)</td>
</tr>
<tr>
<td>Number of tokens</td>
<td>225001</td>
<td>12355</td>
<td>12418</td>
</tr>
<tr>
<td>Number of frames</td>
<td>540681</td>
<td>29686</td>
<td>29918</td>
</tr>
<tr>
<td>Mean frames per utterance</td>
<td>16.0</td>
<td>15.8</td>
<td>16.0</td>
</tr>
<tr>
<td>Out-of-vocabulary rate</td>
<td>1.99%</td>
<td>2.42%</td>
<td>2.79%</td>
</tr>
</tbody>
</table>

Table 1: Statistics of SAYCam-S.

were obtained by linearly interpolating the time span of the original transcript. We filter the utterances by excluding child-produced utterances, retaining only those from parents. For each utterance, we extract multiple frames at 5 frames per second (fps) from the video, up to the first 6.4s of its time span.

The dataset is randomly split into training, validation and test sets (90%/5%/5% of all utterances, respectively). In this work, only the training and validation sets are used, while the test set is left for future use. Our vocabulary is built from all tokens contained in the training set, excluding those with a frequency less than 3 in this set, resulting in a final vocabulary size of 2,350. Any out-of-vocabulary tokens are replaced by the special \texttt{<UNK>} token. Appendix A.1 contains additional details.

The preprocessed dataset consists of 37,486 child-directed utterances (249,774 tokens) paired with 600,285 image frames. Table 1 contains further descriptive statistics about the dataset, and Figure 1 shows some sample frames from the dataset paired with their corresponding utterances. Notably, the average utterance length is rather short compared to sentence lengths in typical written corpora, which is a characteristic of child-directed speech.

3 Neural Networks and Training

3.1 Language-only networks

We use two kinds of networks to encode the language input: single-layer uni-directional LSTM (Hochreiter and Schmidhuber, 1997), which is a variant of Recurrent Neural Network (RNN), and CBOW (Mikolov et al., 2013). Their training objective is token prediction in context using a cross-entropy loss.

Figure 2(b) illustrates the architecture of a uni-directional LSTM. A uni-directional LSTM processes a sequence of tokens left-to-right, and maintains a hidden state after each step, keeping track of context using only tokens to the left of the predicted token in the utterance. The dimensions of the hidden states and the word embeddings are both 512. When predicting the next token, the LSTM assigns a probability distribution over all tokens in the vocabulary.

Figure 2(a) illustrates the CBOW architecture. For CBOW, the context it can see is a constant number of tokens to the left and right of the predicted token. The set of these tokens are called its “context window”. One advantage this provides over uni-directional networks is that the CBOW can additionally utilize information from the right of the token to be predicted. However, unlike the LSTM, its context window size is fixed to a small number, preventing it from modeling long-distance dependencies. CBOW also has a simpler architecture compared to the LSTM: it uses an embedding layer to first embed the discrete input tokens into their word embeddings. Then, all word embeddings within the context window are averaged and then projected by an output layer, producing the predicted distribution over all tokens. All embeddings are of size 512. See Appendix A.2 for additional details regarding network architectures and training configurations.

2 Although this interpolation procedure did not lead to time spans that were exactly aligned with each of the spoken utterances, the relative stability of visual information across seconds meant that the approximate alignment was still informative. We note that noise introduced at this step would lead to an underestimate, not an overestimate, of learnability.

3 The temporal order of utterances is not taken into account. They are also randomly ordered when presented to the network. So the network treats each frame-utterance pair as an independent datapoint.

4 The hidden state and embedding sizes were not critical for our analyses; Smaller embedding dimensions led to degradation of performance on token prediction, but the qualitative conclusions of our analyses remained unchanged.
you like bananas?  
a little banana there.  
here's your water.  
here's a little bit of egg.  
you go bread...  
there's a rice biscuit for you.

ok, see the ball?  
there's the ball.  
where's the ball?  
ouch.  
where's the ball?  
is that the ball?

and here is a farm with a cow on it.  
and the cow has an udder; and then milk comes out of the udder.  
with out hands yeah.  
do you want to go back to the farm sometime?  
yeah we might go this weekend sometime to the farm again.  
and then we have to pour the milk, you pour the milk that last buckets into a big milk truck.

Figure 1: Example frames and their corresponding utterances. Each row is a different scene: a meal at breakfast, a game with a ball, and reading a farm-themed picture book. Unlike common image-text datasets in machine learning, the utterances only loosely align to the frames. For instance, the foods mentioned in the utterance are not always in the corresponding video frames, and the ball mentioned in the utterance is sometimes covered by the cup.

A: CBOW

B: LSTM

C: Captioning LSTM

Figure 2: The three neural network architectures. (a) The CBOW network predicts a missing word given a surrounding context of fixed size. The LSTM (b) and Captioning LSTM (c) networks both predict the next word given a sequence of previous words (additionally a corresponding image for the Captioning LSTM). The light blue boxes indicate word embeddings, the dark blue boxes indicate hidden embeddings, and the red box indicates the visual embedding. Figure adapted from Lake and Murphy (2021).

We measure these networks’ performance on token prediction by per-token perplexity.\(^5\) Our LSTM and CBOW models reached an average perplexity of 24.80 (SD = 0.21) and 22.20

\(^5\)In natural language processing, perplexity is a measure of how well a predicted distribution matches the ground-truth one-hot token distribution, defined as \(\frac{1}{p(y)}\), where \(p(y)\) is the predicted
SD = 0.01) on the validation set, respectively, averaged over 3 runs with different random seeds. CBOW is marginally better than the LSTM on this measure, although it has the benefit of incorporating bidirectional context. For CBOW, we tested context window sizes ranging between 1 to 4 tokens on both sides of the predicted token and found that a context window containing only 1 token on both sides performed best.\(^7\)

### 3.2 Multimodal network

Another advantage of SAYCam-S is its multimodality: it contains parallel vision and language inputs. Adding visual information provides grounding for words, potentially allowing the networks to learn references from words to objects, or at least visual features in the input (Hill et al., 2021; Vong and Lake, 2022). Multimodal learning has been shown to help resolve ambiguities when only linguistic information is present (Berzak et al., 2015; Christie et al., 2016), induce constituent structures (Shi et al., 2019), and ground events described in language to video (Siddharth et al., 2014; Yu et al., 2015).

As a way to incorporate the aligned visual modality for in-context token prediction, we treat each utterance as the caption of its associated frames. We then build an image captioning network (Xu et al., 2015), which is a uni-directional LSTM with the same architecture as described above, with an additional capacity to process information from visual inputs. This Captioning LSTM architecture is illustrated in Figure 2(c). We use a Convolutional Neural Network (CNN) as our vision encoder (specifically, ResNeXt-50 32x4d; Xie et al., 2017), pretrained via unsupervised learning from the visual stream of child S (the single child we focus on) in SAYCam (Orhan et al., 2020). The visual representation produced by the vision encoder is used to initialize the hidden state of the uni-directional LSTM. See Appendix A.2 for additional details.

The perplexity of our Captioning LSTM was 22.10 \((SD = 0.20)\) averaged over 3 runs, which was incrementally lower than the language-only LSTM, suggesting a minor benefit of information from the additional visual modality. Noise in the alignment between the visual and language streams likely damped the size of the improvement. We discuss this issue further in the context of the limitations of the multimodal objective in the General Discussion.

### 4 Results

#### 4.1 Learning from language only

##### 4.1.1 Syntactic and semantic categories

Following Elman (1990), we performed several analyses to assess the syntactic and semantic category structures acquired by the networks. In his simulations, Elman used artificial data consisting of simple, short sentences that were generated from manually designed templates. The templates were filled with words which were explicitly organized in a hierarchical category structure. The representations learned by a SRN trained on this data were visualized using cluster dendrograms of hidden unit activation vectors. The dendrograms showed the emergence of soft, hierarchical category structures of words: two large categories for nouns and verbs, and finer subcategories for each of them, including animate and inanimate nouns in the noun category, and transitive and intransitive verbs in the verb category. Though this successfully demonstrated the network’s ability to learn category structures from distributional cues, the learning setup was synthetic and controlled.

Here, we find neural networks trained on SAYCam-S can learn similar hierarchical syntactic and semantic category structures over the vocabulary, across three separate analyses. We report the results for the LSTM in the main text and the corresponding results for CBOW can be found in Appendix A.3. First, as in Elman (1990)’s SRN, we find that representations learned by the LSTM and CBOW form clusters corresponding to syntactic categories, including nouns and verbs. The verbs also form finer subcategories including transitive and intransitive verbs. These findings are shown in Figure 3; we visualize the LSTM’s word embeddings using t-SNE (van der Maaten and Hinton, 2008) and a dendrogram for the most frequent 24 probability of the ground-truth token \(y\). For a corpus consisting of \(n\) tokens, the perplexity is defined as \(\exp\left(\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}(y_i)\right)\), where \(y_i\) is the \(i\)-th token. The lower the perplexity, the better.\(^6\)

\(^6\) In order to make perplexity as comparable as possible across LSTM and CBOW, all these numbers exclude Start-Of-Sequence (SOS) and End-Of-Sequence (EOS) tokens appended to the starts and ends of utterances, so they are evaluated on the same set of tokens.

\(^7\) Note that it has been shown that small contexts primarily encode syntactic aspects over thematic ones (Chang and Deák, 2020; Huebner and Willits, 2018).
Figure 3: Clustering LSTM’s word embeddings for syntactic categories. For two embeddings $u, v$, t-SNE uses $1 - \cos(u, v)$ as the distance metric, and dendrogram uses $\cos(u, v)$ as the similarity measure. Nouns and verbs form two large clusters. Transitive and intransitive verbs form two smaller subclusters.

nouns—12 transitive verbs, and 12 intransitive verbs that are unambiguous in their transitivity (see Figure 8 in the Appendix for CBOW results). Both the t-SNE and dendrogram use cosine-based metrics between word embeddings.\(^8\) Furthermore, Figures 10 and 11 demonstrate that clusters for other syntactic categories like adjectives and adverbs also emerge from training.

Second, we find that the representations learned by the LSTM forms clusters corresponding to semantic subcategories of nouns. We manually label the most frequent nouns that are unambiguously in different semantic categories, using a reference set of semantic categories derived from WordBank (Frank et al., 2016). We exclude categories having less than 6 unambiguous words from our analysis. As can be seen from Figure 4, there are several visually identifiable clusters that correspond to different semantic categories.\(^9\) Note that while Elman (1990) found a clear animate versus inanimate distinction among nouns, we did not find such a salient distinction (see Figure 12 in Appendix). Interestingly, some thematically related words (“milk”, “farm”, and “cow”) are close to each other. We find that this cluster can be directly traced back to a particular scene in the training data; these words co-occur in a scene where the parent is reading a farm-themed picture book, illustrated in the third row of Figure 1.

Third, as pointed out by Linzen and Baroni (2021), information in the representation may not be used by the network to causally affect its behavior. We therefore apply additional behavioral tests to provide further evidence for syntactic category structures in our networks. We design a novel cloze test (Taylor, 1953) to evaluate the noun-verb distinction. We build clozes such as “we are going to ___ here”, where the cloze expects either a noun or a verb.\(^10\) Trials are generated by iterating over utterances in the validation set, identifying each token that is a noun or verb, and replacing one of these tokens with an empty slot to create a cloze. For each cloze, we fill the slot with every possible noun or verb in the vocabulary, scoring each

\(^8\)While we used word embeddings to conduct these analyses, mean hidden vectors across the dataset (approach used by Elman 1990) yield similar results.

\(^9\)CBOV results are shown in Figure 8 in the Appendix; there are some identifiable clusters but they are less clear than clusters from the LSTM.

\(^10\)This approach is similar to the category distinction test for masked language models in Kim and Smolensky (2021).
Figure 4: Clustering LSTM’s word embeddings for semantic categories. Again, both plots use cosine measures in Figure 3. We present the most frequent 6 words from 8 different categories. Most distinct clusters clearly correspond to semantic categories.

<table>
<thead>
<tr>
<th>Top-5 predictions</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>we should turn on some lights, huh?</td>
<td>91.2% put 5.2% turn 0.4% leave 0.4% keep 0.4% get</td>
</tr>
<tr>
<td>we should turn on some lights, huh?</td>
<td>14.0% lights 13.4% toys 9.5% water 7.6% music 5.4% books</td>
</tr>
<tr>
<td>are you done going potty?</td>
<td>9.3% done 6.4% ‘re 6.0% feeling 5.5% hiding 5.4% are</td>
</tr>
<tr>
<td>and there’s a kitty looking at a mouse.</td>
<td>40.9% kitty 18.9% mouse 4.3% doggy 3.8% door 2.3% dog</td>
</tr>
<tr>
<td>we might go to the beach today.</td>
<td>61.2% library 10.1% playground 8.8% beach 2.9% park 2.9% farm</td>
</tr>
<tr>
<td>now on our way we can get some food for us for breakfast</td>
<td>56.2% bread 6.9% chicken 4.2% strawberries 4.0% water 3.9% salmon</td>
</tr>
<tr>
<td></td>
<td>37.0% lunch 11.6% breaky 11.4% dinner 6.9% oil 6.0% clothes</td>
</tr>
</tbody>
</table>

Table 2: Examples of clozes and the networks’ predictions. We present a cloze by underlining the ground-truth word at the slot. We list the top-5 predictions in this form: (predicted normalized probability, word). The top predictions frequently align with expected categories. For instance, a noun follows a determiner, and a word in the food-drink category occurs if breakfast is mentioned. By comparing the predictions of the LSTM and the CBOW, we can also see the disadvantages of CBOW’s small context window.
Figure 5: Mean syntactic probing accuracy over subsets for each network and syntactic phenomenon. The top group is the mean over all subsets, and each of the rest groups is the mean over the subsets for this phenomenon. Each label for a phenomenon is accompanied by an illustrative example, in which the first option in the bracket is grammatical, while the second is not. The model is correct if it assigns a higher probability to the grammatical sentence over the ungrammatical one. The dashed line denotes chance accuracy. See Appendix A.5 for fine-grained results on each phenomenon.

particular slot. Across the 2412 clozes we generated (with a base rate of 65% verbs), LSTM achieves a high accuracy of 97.96% ($SD = 0.23\%$ over 3 runs) and CBOW achieves an accuracy of 91.20% ($SD = 0.33\%$). Table 2 presents some cloze examples and top predictions from our networks. Appendix A.4 contains more details regarding cloze construction and additional examples. Overall, these results demonstrate the network’s ability to contextually differentiate nouns and verbs, supplementing our earlier findings.

4.1.2 Linguistic Acceptability Analysis

Next, we examine the networks’ sensitivity to acceptability of a sequence modulated by more complex linguistic phenomena such as subject-verb agreement and argument structure, again following Elman’s lead (1989; 1991). We study this using Zorro: a minimal pair test suite for 13 different linguistic phenomena (Zorro; Huebner et al., 2021), which itself is derived from another minimal pair test suite (BLiMP; Warstadt et al., 2020). The minimal pair approach asks models to judge which of two sentences is more acceptable (e.g., “I saw this toy” vs. “I saw this toys”). The sentences in a minimal pair highlight a single linguistic phenomenon that leads to a contrast in acceptability judgments. We filter the Zorro dataset such that only sentence pairs that are entirely within our models’ vocabulary are included. This leaves us with 15 subsets of the dataset, corresponding to 7 different linguistic phenomena; 8 were excluded for having no items after filtering. Additional details regarding dataset curation can be found in Appendix A.5.

On these filtered subsets, we test and compare several networks: the three networks we trained (language-only LSTM, CBOW, and Captioning LSTM\(^\text{11}\)), two baseline n-gram language

\(^\text{11}\)The Captioning LSTM always needs an image input, so we used the mean image frame of the training set in this evaluation. As shown in Figure 5, its performance is not substantially different from the language-only LSTM.
models based on statistics of the training set (unigram and bigram language models\textsuperscript{12}), and a strong Transformer model (pre-trained weights from Huebner et al. 2021 trained on AO-CHILDES which aggregates data from many children). The results are summarized in Figure 5. Though the networks trained on SAYCam-S perform worse than the Transformer trained on more data, they are clearly above chance on many tests. For example, the LSTM achieves 67.7\% accuracy on determiner-noun agreement, and the CBOW achieves 61.1\% accuracy. The lower performance of CBOW on this test can be explained by the length of the dependency that needs to be processed. That is, some of the dependencies in this test span longer distances than CBOW’s context window, which is advantageous for the LSTM. However, on the subject-verb agreement test which requires even longer dependencies, the LSTM does not perform substantially above chance (55.7\%). It is possible that there are too few distributional cues for long-distance agreements in SAYCam-S in particular, although other findings have shown that RNNs (Elman, 1991; Linzen and Leonard, 2018) and Transformers (Tay et al., 2021; Pérez-Mayos et al., 2021) with modest amounts of training data in general have increased difficulty with longer-distance dependencies.\textsuperscript{13} Other tests such as quantifiers and grammatical case are less useful for distinguishing between models because the unigram and bigram models performed well, indicating that even very simple distributional statistics are sufficient for high accuracy on these tests.

4.2 Learning from multimodal input
As mentioned earlier, the LSTM showed an incremental improvement in perplexity with additional visual information. In this final set of analyses, we examine how incorporating visual information influences the linguistic representations in the Captioning LSTM.

4.2.1 Sources of multimodal improvement
To investigate the areas of possible improvement, we first measure the improvement in cross-entropy loss for words occurring at least twice in the validation set, grouped by each word’s syntactic category. This difference in loss between the Captioning LSTM and the language-only LSTM is shown in Figure 6. The improvements for most syntactic categories are statistically significant (Table 6 in Appendix), but in particular, nouns and verbs benefit the most from additional visual information. The improvement for nouns is expected, since most nouns acquired early by children can be visually grounded (Frank et al., 2021). Surprisingly, verbs and even function words show some improvement, even though they are often more challenging to directly ground in images.

It is challenging to discern precisely which visual-linguistic correlations are responsible for the improved predictive power. Nevertheless, in Figure 7, we provide several examples and compare the cross-entropy losses of the text-only LSTM and Captioning LSTM on each token of the utterances. For concrete nouns like “ball” in the third example, introducing frames containing clear referents greatly reduces losses on them. In other examples, however, the influence of visual information is not clearly beneficial or interpretable. For example, on the

\textsuperscript{12}N-gram models are simple language models based on token counts in a corpus. An n-gram is n consecutive tokens. The unigram model is based on counts of individual token, without considering any context. The bigram model is based on counts of token pairs occurring together, and so on. We tried larger n-gram models for the acceptability analysis, but they performed similarly to the bigram model due to data sparsity and their back-off mechanism.

\textsuperscript{13}In fact, the AO-CHILDES Transformer trained on more data also shows comparatively worse performance on this test compared to other tests.
Figure 7: Predicting an utterance with (Capt. LSTM) and without (LSTM) access to a video frame. The numbers above each token show the models’ losses when predicting particular tokens (heatmap normalized within an utterance). The mean loss \( M \) is also shown. The Captioning (Capt.) LSTM has better mean loss than the LSTM on all examples, and the word predictions for some visible objects are improved over the LSTM (“doggy”, “ball” in third row, etc.). The last three examples are harder to interpret: the Capt. LSTM fails to make better word predictions for other visible objects (“ball” in fourth row and “car”).

Fourth example, the loss on “car” decreased, but the loss on “ball” increased despite both referents being present in the frame. This suggests the network also acquires less interpretable and indirect visual-linguistic correlations. One possible hypothesis for the additional improvements in cases where there are no direct referents in the scene is that different visual moments in childhood (e.g. mealtime vs. play) elicit sufficiently different distributions of words (Roy et al., 2015), although we leave further investigation in this direction for future work.

4.2.2 Influence on representations

As a second analysis on how visual information influences linguistic representations, we perform Representational Similarity Analysis (RSA; Kriegeskorte et al., 2008) across the three neural networks. We compute the dissimilarity matrices of the three networks’ representations for the set of words in the aforementioned syntactic category analysis in Section 4.1.1, using a dissimilarity metric: \( \frac{1}{2}(1 - \cos(u, v)) \). Visualizations of these matrices can be found in Appendix A.6.

The similarity between representations of two networks is the Pearson correlation between elements in the upper triangles of their dissimilarity matrices. The two networks based on the same LSTM architecture (language-only LSTM and Captioning LSTM) are quite similar to each other (\( r(1126) = .82, p < .001 \)), while CBOW is less similar to either LSTM (\( r(1126) = .71, p < .001 \) to LSTM, \( r(1126) = .70, p < .001 \) to Captioning LSTM). The high similarity between the LSTM and Captioning LSTM is consistent with recent studies which
found that incorporating visual information does not dramatically restructure or improve linguistic representations (Iki and Aizawa, 2021; Yun et al., 2021).

5 General Discussion

Our work demonstrates what kinds of linguistic knowledge are learnable from the naturalistic input received by a single child. There are three main takeaways. First, using the SAYCam dataset (Sullivan et al., 2021) and techniques from modern machine learning and natural language processing, we find that neural networks learning exclusively from developmentally plausible data can differentiate words in different syntactic categories. These categories help to shape the networks' behaviors, in predicting a token's category based on context and in acquiring sensitivity to phenomena such as determiner-noun agreement, although longer distance dependencies proved more difficult (e.g., subject-verb agreement). Second, the networks can also organize nouns into semantic categories such as animals, body parts, and clothing, largely following a taxonomic organization mixed with some thematic influences. Finally, we found that introducing visual information brings an incremental improvement for predicting words in context, with relatively larger improvements for syntactic categories such as nouns and verbs. However, the acquired linguistic representations in the LSTMs were similar regardless of whether it received visual information.

A distinguishing aspect of our work is using naturalistic, multimodal data from a single child. Elman's pioneering work (1989; 1990; 1991) showed how Simple Recurrent Networks (SRNs) can learn meaningful syntactic and semantic representations without targeted inductive biases. The NLP community has continued this tradition, using modern successors of the SRN for modeling sequences (LSTMs, Transformers, etc.) trained on larger-scale written text corpora (Belinkov and Glass, 2019; Rogers et al., 2021; Linzen and Baroni, 2021; Warstadt and Bowman, 2022). Moreover, neither synthetic nor written text is essential: networks can also learn useful syntactic and semantic representations when trained on the naturalistic, noisy data received by multiple children (Huebner and Willits, 2018; Huebner et al., 2021; Fourtassi, 2020). Our work takes a further step in demonstrating how the same types of regularities, although in more nascent forms, emerge from neural networks trained on the linguistic input received by just one child. Furthermore, we also provide an initial examination of what additionally can be learned when visual data is paired with the linguistic input, complementing previous work training vision-only models on SAYCam (Orhan et al., 2020; Zhuang et al., 2021).

Although we focused on the outcome of learning rather than the stages of learning—that is, we did not seek to build a developmental cognitive model—it is still instructive to compare our findings to studies of language acquisition in children. We have demonstrated that distributional information in the input to a child before 25 months of age is enough to support the formation of syntactic categories, including nouns and non-alternating transitive and intransitive verbs. Meanwhile, children's category structures develop at varying pace. For example, children at around 23 months can productively use novel nouns but not verbs, indicating the formation of noun category but not verb category at this age (Tomasello and Olguin, 1993; Olguin and Tomasello, 1993). Our networks' failure to acquire more complex linguistic phenomena, in particular subject-verb agreement, may also benefit from a parallel discussion with developmental work. English-speaking children have been reported to successfully produce subject-verb agreement markers between the ages of 2;2 and 3;10 (Brown, 1973). Given that the endpoint of our training data is 25 months, it may be the case that access to a child's linguistic input that extends beyond this timeframe is required. Furthermore, the comprehension of subject-verb agreement has been known to be delayed in English-speaking children (Johnson et al., 2005; Legendre et al., 2014), and Legendre et al. (2014) in particular proposed a hypothesis that attributes this delay to reliability of distributional cues in the input. In this regard, our results provide a piece of supporting evidence speaking to the weakness of distributional cues for subject-verb agreement in early child-directed input. Regarding semantic development, our results showed that the emergent semantic clusters correspond to real superordinate categories that children learn (“animal”, “vehicle”, etc.), although exactly when and how children learn these concepts is still a puzzle (Murphy, 2002). Somewhat counter-intuitively, infants can typically discriminate between visual exemplars of broader categories (animal vs. vehicle) before more specific ones (Saint Bernard vs. Beagle) (Mandler and McDonough, 1993; Quinn, 2004), yet the corresponding words for superordinate categories are acquired comparatively late relative to basic-level words (Murphy, 2002). Multimodal
models trained on SAYCam could potentially provide a unique lens into these questions, although more work is needed.

Our work only scratches the surface of understanding what is learnable from a young child’s experiences. SAYCam offers an unprecedented snapshot of three children’s experiences, but it captures only a small fraction of their total linguistic input, preventing us from training larger and more sophisticated networks (e.g., Transformers; Vaswani et al., 2017) or analyzing more complex linguistic phenomena (Belinkov and Glass, 2019; Rogers et al., 2021; Linzen and Baroni, 2021). The challenges of training multimodal models are particularly noteworthy. Beyond imperfections in pre-processing (Section 2) and inherent stochasticity in a child’s gaze (Yu et al., 2021), using text rather than audio removes phonological or morphological cues, while also treating segmentation capabilities as given (Meylan and Bergelson, 2022).

Moreover, we did not fully incorporate the temporal nature of a child’s experience, both in how the videos were converted to still images (impeding learning of certain kinds of words that might require visuotemporal integration, e.g. “pick” and “take”; Ebert and Pavlick, 2020) and how networks were trained on the whole corpus simultaneously (one alternative, training networks on age-ordered data, can be found in Huebner and Willits, 2020). Finally, and perhaps most importantly, the networks must learn passively from a child’s fundamentally active and embodied experiences. The networks cannot choose their own actions to take in the environment, do not have desires and goals, and do not realize that language can be a means of achieving what they want. In all of these ways, the types of neural networks considered here, even when scaled up, are far from understanding language in all the ways that people do (Lake and Murphy, 2021). Nevertheless, our results show that neural networks can acquire meaningful structure from a real snapshot of developmental experience. Stronger models, paired with denser and higher-resolution developmental snapshots, would undoubtedly lead to further discoveries.

Acknowledgments
We are grateful for Jeffrey Elman and his work on “Finding structure in time” that continues to guide the fields of cognitive science and natural language processing over 30 years later. We are also grateful for the volunteers who contributed to the SAYCam dataset (Sullivan et al., 2021) that made our article possible.

References


A Appendix

A.1 Dataset Details

SAYCam (Sullivan et al., 2021) is a longitudinal dataset consisting of egocentric head-mounted camera recordings from 3 children (S, A and Y), whose recordings span the ages 6–30, 8–31, and 7–24 months respectively. Recordings took place for a few hours each week over this course, amounting to 100–200 hours of recorded video data per child and more than 415 hours in total. As mentioned in the main text, we only use the data from baby S since their videos had the largest proportion of speech transcribed. The speech transcribed for this child spans 6–25 month of age. Each transcript contains the relevant information for our purposes, including the utterances, the speaker and the time of the utterance (in seconds).

Preprocessing of transcripts. There was considerable noise in the original transcripts, requiring a number of preprocessing steps before feeding them as input to our networks. Some of these issues included very long annotations of multiple sentences, sometimes spanning minutes of video, and inconsistencies across transcripts. To resolve the first issue regarding long annotations, we use spaCy (Honnibal and Montani, 2017) to split annotated utterances into shorter sentences, which are the utterances we actually use. As we mentioned in the main text, we label the time span of each utterance by linearly interpolating (i.e., evenly segmenting) the time span of the original transcript. We filtered these utterances, retaining only those from either parent, which comprised the majority of the child-directed speech. The second issue is also mitigated by the spaCy tokenizer (Honnibal and Montani, 2017). For example, it separates the “i’m” into “i” and “’m”, and “im” into “i” and “’m”, so that the model can recognize the same “i” across inconsistent transcripts. Of course, this is still imperfect, and we leave further improvements for future work. When presented to the network, out-of-vocabulary tokens in utterances are replaced by <UNK>, and utterance lengths are truncated to at most 25 tokens.

Preprocessing of video frames. The original resolution of video frames from SAYCam are $640 \times 480$. In order to more closely mimic the view from the child and fit the input shape of our pretrained ResNeXt network (Orhan et al., 2020), we first resized the minor edge to 256 and then applied a $224 \times 224$ square crop centered at 16 pixels lower the center of each original frame. For each utterance, we extracted multiple video frames using this procedure at a rate of 5fps starting from the beginning of its time span until reaching the end of the time span or 32 extracted frames (6.4s of video).

A.2 Network and Training Configuration

Network Configuration. For all networks, we use embedding and hidden size 512. For the LSTM and the Captioning LSTM, we tie weights in the word embeddings with weights in the output layer and add bias terms with their output layers. For the CBOW, we do not tie the weights in the input and output embedding matrices, nor do we add bias terms.

For the LSTM, the initial hidden state at the beginning of the sequence is all zeros. For the Captioning LSTM, we add a linear adapter layer on top of the vision encoder to project the visual representation and this projected representation is used to initialize the hidden state of the uni-directional LSTM. We freeze the stem of the vision encoder and only train the adapter and LSTM. When training the network, we randomly sample a frame from the multiple frames aligned with the utterance, applied data augmentation, and yield an example pair (frame, utterance). The data augmentations we applied are the following (in PyTorch):

```python
transforms.Compose([
    transforms.RandomResizedCrop((224, 224), scale=(0.2, 1.)),
    transforms.RandomApply([GaussianBlur([.1, 2.])], p=0.5),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]),
])
```

Note this is the same set of augmentations as in the codebase of Orhan et al. (2020)\textsuperscript{14}, with the exception of ColorJitter as it breaks the correspondence between color words and color in images.

Training Configuration. For training the LSTM and the Captioning LSTM, we use batch size 16, initial learning rate $6 \times 10^{-3}$, and dropout on the input word embeddings with dropout

\textsuperscript{14}https://github.com/eminorhan/baby-vision
rate 0.5. For training the CBOW, we use batch size 8, initial learning rate $3 \times 10^{-3}$, and
dropout on the output embeddings with dropout rate 0.1. For all networks, we use the AdamW
optimizer and apply weight decay of 0.04. For the LSTMs, we apply learning rate scheduling
by reducing the learning rate by a factor of 10 when the validation loss has not improved
across consecutive 5 epochs (same for CBOW, but with a 2 epoch threshold). The loss for a
batch of utterances is the mean cross-entropy across all tokens. We apply early-stopping by
training the network until convergence and selecting the checkpoint with the lowest loss on the
validation set. All the hyperparameters are also tuned toward this validation loss. We trained
each network with 3 different random seeds.

The performance of our networks measured in perplexities is shown in Table 3.

Table 3: Token prediction perplexities of networks on the validation set. In order to make comparison
across uni-directional networks and CBOW, we report perplexities excluding both the SOS and the
EOS token. Numbers are the means of 3 runs, and numbers in the bracket are the standard deviations.

<table>
<thead>
<tr>
<th>Model</th>
<th>perplexity (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBOW</td>
<td>22.20 (0.01)</td>
</tr>
<tr>
<td>LSTM</td>
<td>24.80 (0.21)</td>
</tr>
<tr>
<td>Captioning LSTM</td>
<td>22.10 (0.20)</td>
</tr>
</tbody>
</table>

A.3 Additional Clustering Figures
In this section, we include plots demonstrating that the learned networks are sensitive to other
kinds of syntactic and semantic structure. First, we show additional t-SNE and dendogram
plots for the CBOW network showing that it can also differentiate nouns vs. verbs (Figure 8),
and also different semantic categories (Figure 9). Figures 10 and 11 show t-SNE and dendogram
plots that include words from additional syntactic categories (adjectives and adverbs) for the
LSTM and CBOW networks respectively, showing that both networks from clusters that
correspond to each kind of syntactic category. Finally, Figure 12 presents a t-SNE plot showing
different word embeddings from the LSTM network colored by animacy.

A.4 Cloze Test Details
As described in the main paper, we create clozes from utterances in the validation set. We
filtered out clozes that contained less than two words, or occurred in the training set. This
resulted in an evaluation set containing 2412 clozes, with 848 (35%) for nouns and 1564 (65%)
for verbs. However, we noticed that many clozes have original words that are atypical nouns
and verbs, such as be-verbs, modal verbs, quantifiers, words ambiguous in their part-of-speech,
or <UNK> token. As a robustness check, we re-ran our cloze analysis after filtering out these
clozes. This left 1682 clozes, with 795 (47%) for nouns and 887 (53%) for verbs. On this filtered
set, our language-only LSTMs achieve 97.30% (SD = 0.33%) accuracy, and CBOW achieve
89.22% (SD = 0.23%) accuracy. These accuracies are still high and similar to the results from
the unfiltered set, suggesting that our results are robust to differences in vocabulary.

Additional cloze examples are shown in Table 4, showing the top model predictions for 3
runs of LSTM and CBOW. From these examples you can see networks are clearly forming
word clusters corresponding to interpretable categories, not only larger syntactic categories
like nouns and verbs, but also finer categories like animals and places, and other categories
like be-verbs, words following “an”, ditransitive verbs and V-ings. The LSTM tends to copy
a word from the context, if that fits in the category. Also, the CBOW, which utilizes only
near contexts, is doing surprisingly well, which indicates many unexpected correlations in the
distributional patterns.

A.5 Linguistic Acceptability Analysis
As we mentioned in the main paper, we evaluated our networks on a subset of Zorro (Huebner
et al., 2021), a minimal pair test suite consisting of 13 linguistic phenomena comprised of one or
more subsets. Each subset contains 4,000 sentences making up 2,000 minimal pairs. Sentences
in Zorro were created using templates filled with words from word lists they curated. Their
word lists contained frequent words in the datasets they used. However, the word distribution
in their datasets are different from ours. Among the 646 word types that occurred in Zorro,
only 403 were in our vocabulary; most words not in our vocabulary were either human names,
Figure 8: Clustering CBOW’s word embeddings by cosine measures in Figure 3. Nouns and verbs form two large clusters. Transitive and intransitive verbs form two smaller subclusters.

Figure 9: Clustering CBOW’s word embeddings by cosine measures in Figure 3. The cluster structure is less clear but several still correspond to semantic categories.
Figure 10: Clustering LSTM’s word embeddings by cosine measures in Figure 3. Clusters generally correspond to syntactic categories.

Figure 11: Clustering CBOW’s word embeddings by cosine measures in Figure 3. Clusters generally correspond to syntactic categories.
The set of words here is the same set in Figure 4. Hue means animacy. Our animacy data is from https://osf.io/4t3cu/ contributed by Joshua VanArsdall and Janell Blunt. The animacy shown here is from their AnimPhysical field. For each word in our vocabulary, we try to get its animacy by looking up in the data its base form obtained by NLTK (Bird et al., 2009) lemmatizer; if not found, we do not include this word. Comparing this plot to Figure 4, you can see animacy within a semantic category is relatively uniform. But there is no clear overall animacy structure.

more abstract words usually not present in the early children’s vocabulary (e.g., “control”, “tradition”, “bank”), or different word-forms (e.g., plural, past tense). Therefore, we filtered the sentence pairs so that they only consist of words contained within the vocabulary of our dataset. Table 5 lists the number of sentence pairs left in each subset, showing the remaining linguistic phenomena that we could evaluate our networks on.

The full set of results across each individual subset is shown in Figure 13. The Transformer network performs best on most of the tests. Note also that CBOW and the N-gram models do well on many subsets. This is due to local co-occurrence cues and word frequency effects. The LSTM is better on subsets where longer-distance dependencies are required, such as agreement determiner noun/across 1 adjective and quantifier/existential there. The Captioning LSTM performs mostly close to the language-only LSTM; the only notable difference, shown in Figure 13, is that it is noticeably better on the quantifiers-superlative subset, close to the CBOW.

A.6 Cosine Similarity Heatmaps

Figures 14, 15 and 16 plot the cosine similarity between N words for the LSTM, Captioning LSTM and CBOW models respectively, showing the within and across similarity between different syntactic categories. These similarity matrices are used to calculate the Pearson correlations in the Representational Similarity Analysis in Section 4.2.2.
Figure 13: Accuracy on linguistic acceptability tests. The dashed line means the chance level.
Figure 14: Heatmap of cosine similarity of LSTM’s word embeddings. Nouns and verbs are more similar to other words within the same category than other words in the different category.

Figure 15: Heatmap of cosine similarity of Captioning LSTM’s word embeddings. Nouns and verbs are more similar to other words within the same category than other words in the different category.
Figure 16: Heatmap of cosine similarity of CBOW’s word embeddings. Nouns and verbs are more similar to other words within the same category than other words in the different category.

Figure 17: Heatmap of cosine similarity of LSTM’s word embeddings. Words are more similar to other words within the same category than other words in the different category.
Figure 18: Heatmap of cosine similarity of Captioning LSTM’s word embeddings. Words are more similar to other words within the same category than other words in the different category.

Figure 19: Heatmap of cosine similarity of CBOW’s word embeddings. Words are more similar to other words within the same category than other words in the different category.
Model Top-5 predictions

that ’s an o!
LSTM 36.7% egg 8.1% emu 4.4% eagle 4.2% q 3.6% ant
LSTM 44.9% emu 6.6% echidna 6.2% ant 4.3% egg 2.3% s.
LSTM 25.2% emu 14.3% egg 9.5% echidna 6.8% ant 5.1% q
CROW 64.1% hour 19.2% ant 4.1% apple 3.8% emu 2.6% egg
CROW 53.3% hour 16.7% ant 8.1% apple 6.6% emu 5.3% egg
CROW 49.0% hour 20.9% ant 7.6% apple 6.9% emu 5.6% egg

theres a strawberry and theres a flower
LSTM 69.7% s 17.7% ’s 8.0% is 1.8% was 1.4% ’s
LSTM 79.4% s 10.6% is 8.9% ’s 0.4% are 0.2% was
LSTM 73.2% s 16.7% ’s 7.7% is 0.9% ’s 0.8% was
CROW 57.9% ’s 20.7% s 20.5% is 0.5% was 0.2% are
CROW 57.9% ’s 20.8% s 20.4% is 0.4% was 0.2% are
CROW 57.6% ’s 21.0% is 20.5% s 0.4% was 0.2% are

theres a strawberry and theres a flower
LSTM 26.6% leaf 9.1% car 9.0% cardigan 7.0% cupcake 5.4% flower
LSTM 15.2% ball 14.0% strawberry 7.6% bear 6.6% banana 6.1% kitty
LSTM 9.3% cupcake 8.3% kitty 5.5% cup 5.3% leaf 5.3% cardigan
CROW 9.2% magazine 7.4% bug 7.3% moment 7.0% biscuit 6.5% horse
CROW 10.0% magazine 8.5% bug 6.9% moment 6.4% biscuit 5.9% horse
CROW 9.7% magazine 8.3% moment 6.6% bug 6.5% biscuit 5.9% horse

can you show me the eggs?
LSTM 33.4% give 25.9% show 11.3% tell 6.3% pick 4.3% get
LSTM 63.8% show 21.2% give 7.1% get 1.7% find 1.5% throw
LSTM 56.2% show 37.0% give 1.8% get 1.7% throw 0.4% lift
CROW 61.5% show 16.9% give 11.2% want 6.2% tell 1.5% showing
CROW 62.3% show 16.6% give 11.2% want 5.8% tell 1.7% showing
CROW 63.2% show 16.3% give 11.0% want 5.3% tell 1.6% showing

you keep eating.
LSTM 36.2% going 28.7% trying 6.0% eating 3.6% done 2.8% holding
LSTM 33.4% going 11.2% done 8.8% eating 8.2% trying 2.0% doing
LSTM 24.2% going 7.8% looking 7.1% doing 5.6% eating 4.4% one
CROW 65.6% going 14.4% eating 4.6% doing 4.6% holding 2.4% trying
CROW 70.8% going 7.9% eating 5.6% holding 3.5% pressing 2.8% trying
CROW 69.2% going 10.4% eating 4.5% trying 3.0% doing 2.6% pressing

Table 4: Additional examples of clozes and the networks’ predictions. Three rows of a same architecture are results from three runs.
Table 5: Number of sentence pairs left in each subset in Zorro. Each subset originally contained 2000 sentence pairs. After filtering, 15 out of 23 subsets, or 7 out of 13 phenomena, have sentence pairs left.

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Subset</th>
<th>#sentence pairs left</th>
</tr>
</thead>
<tbody>
<tr>
<td>agreement determiner noun</td>
<td>across 1 adjective</td>
<td>656</td>
</tr>
<tr>
<td></td>
<td>between neighbors</td>
<td>616</td>
</tr>
<tr>
<td>agreement subject verb</td>
<td>across prepositional phrase</td>
<td>480</td>
</tr>
<tr>
<td></td>
<td>across relative clause</td>
<td>532</td>
</tr>
<tr>
<td></td>
<td>in question with aux</td>
<td>280</td>
</tr>
<tr>
<td></td>
<td>in simple question</td>
<td>836</td>
</tr>
<tr>
<td>anaphor agreement</td>
<td>pronoun gender</td>
<td>0</td>
</tr>
<tr>
<td>argument structure</td>
<td>dropped argument</td>
<td>341</td>
</tr>
<tr>
<td></td>
<td>swapped arguments</td>
<td>529</td>
</tr>
<tr>
<td></td>
<td>transitive</td>
<td>384</td>
</tr>
<tr>
<td>binding</td>
<td>principle a</td>
<td>0</td>
</tr>
<tr>
<td>case</td>
<td>subjective pronoun</td>
<td>527</td>
</tr>
<tr>
<td>ellipsis</td>
<td>n-bar</td>
<td>0</td>
</tr>
<tr>
<td>filler-gap</td>
<td>wh-question object</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>wh-question subject</td>
<td>0</td>
</tr>
<tr>
<td>irregular</td>
<td>verb</td>
<td>0</td>
</tr>
<tr>
<td>island-effects</td>
<td>adjunct island</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>coordinate structure constraint</td>
<td>0</td>
</tr>
<tr>
<td>local attractor</td>
<td>in question with aux</td>
<td>480</td>
</tr>
<tr>
<td>npi licensing</td>
<td>matrix question</td>
<td>374</td>
</tr>
<tr>
<td></td>
<td>only npi licensor</td>
<td>205</td>
</tr>
<tr>
<td>quantifiers</td>
<td>existential there</td>
<td>181</td>
</tr>
<tr>
<td></td>
<td>superlative</td>
<td>188</td>
</tr>
</tbody>
</table>

Table 6: Type level mean loss difference from language-only LSTM to Captioning LSTM on validation set, with t-test results. Results on adjective and cardinal number is not significant.

<table>
<thead>
<tr>
<th>Syntactic Category</th>
<th>#Types</th>
<th>Mean Loss Difference</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>617</td>
<td>-0.31</td>
<td>-11.42</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>noun</td>
<td>220</td>
<td>-0.51</td>
<td>-9.44</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>verb</td>
<td>150</td>
<td>-0.29</td>
<td>-5.58</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>adjective</td>
<td>44</td>
<td>-0.14</td>
<td>-1.75</td>
<td>0.09</td>
</tr>
<tr>
<td>adverb</td>
<td>45</td>
<td>-0.14</td>
<td>-2.23</td>
<td>0.03</td>
</tr>
<tr>
<td>function word</td>
<td>82</td>
<td>-0.13</td>
<td>-3.25</td>
<td>0.002</td>
</tr>
<tr>
<td>cardinal number</td>
<td>11</td>
<td>-0.03</td>
<td>-0.18</td>
<td>0.86</td>
</tr>
<tr>
<td>.</td>
<td>65</td>
<td>-0.16</td>
<td>-2.09</td>
<td>0.04</td>
</tr>
</tbody>
</table>