Appendix: Finding Structure in One Child's Linguistic Experience

Wentao Wang,[†] **Wai Keen Vong**,[†]
Najoung Kim,[†,δ] and **Brenden M. Lake**[†]
[†]Center for Data Science, New York University
[δ]Department of Linguistics, Boston University

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 Section 2, 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 Section 2, 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. As we also mentioned in Section 2, we excluded child-produced utterances to focus solely on the data a child receives, meeting our goal of investigating what can be learned from the input to a child. Additionally, many of the transcribed child utterances, especially earlier in language development, are not very informative. Utterances from people other than the parents are rare. 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. All transcripts were lowercased. 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 640×480. 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 224×224 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). (We wanted to pick a reasonable number of temporal frames that were a power of 2 and where the visual content was mostly similar within the time span. 5fps was based on how Orhan et al. (2020) sampled the frames for their self-supervised training.)

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 starting hidden and cell states at the beginning of the sequence are initialized to 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 and cell states 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):

```
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)[18], with the exception of ColorJitter as it breaks the correspondence between color words and color in images.

Training Configuration. Weights of all networks are randomly initialized by the default setting of PyTorch. Specifically, the weights of the LSTM and the output layer of the CBOW are initialized from Uniform$(-\sqrt{1/d}, \sqrt{1/d})$ where d is the dimension 512, and the weights of the embeddings are initialized from Normal$(0, 1)$. For training the LSTM and the Captioning LSTM, we use batch size 16, initial learning rate 6×10^{-3}, and dropout on the input word embeddings with dropout rate 0.5. For training the CBOW, we use batch size 8, initial learning rate 3×10^{-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 threhold). 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 and the number of trained epochs for each selected best checkpoints are shown in Table 3.

Model	perplexity (SD)	number of trained epochs
CBOW	22.20 (0.01)	{31, 65, 58}
LSTM	24.80 (0.21)	{29, 38, 28}
Captioning LSTM	22.10 (0.20)	{29, 42, 38}

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

A.3 Selection and Categorization of Visualized Words

In this section, we describe the process of selecting and categorizing the words (nouns and verbs) that are visualized in our figures for syntactic (e.g., Figure 3) and semantic (e.g., Figure 4) categories. We first prepared semantic categories of nouns and syntactic categories of verbs: For nouns, we considered semantic categories from WordBank (Frank et al., 2016): sounds, animals, vehicles, toys, food&drink, clothing, body parts, household, furniture&rooms, outside, places, people, games&routines; for verbs, we considered these syntactic categories: transitive, ditransitive, intransitive, transitive/intransitive (which means the verb can be either transitive or intransitive), special (special verbs including be-verbs and modal verbs). We went through the most frequent words in our vocabulary by sorting them in descending order of their frequencies in the training set, and stopped at words with frequency 24 (due to limited time and labor). For each word we encountered, we did our best to classify it into the proper category, using examples from the dataset when needed. We excluded words having any of the following properties: 1) not a common word in the category (e.g., "marmite", "sam"), 2) ambiguous in its

[18] `https://github.com/eminorhan/baby-vision`

category (e.g., "chicken", "breaky", "painting"), and 3) referring to the categories themselves (e.g., "animals", "toys", "food"). For semantic categories, we decided to exclude 5 semantic subcategories (sounds, furniture&rooms, outside, people, games&routines) because we found they do not form coherent category structures or they did not contain enough words. For syntactic categories of verbs, we decided to only include transitive and intransitive verbs. In Table 4, we list samples of words that we excluded following the process described above.

POS	Category	Excluded Words
noun	sounds*	boop bloop ruff ya blo mmkay bop nom quack vroom boom
	animals	marmite chicken animals
	vehicles	N/A
	toys	toys toy book books bubbles dummy marker pen
	food & drink	food breakfast breaky chicken
	clothing	clothes nappy backpack blanket
	body parts	N/A
	household	N/A
	furniture & rooms*	bin potty chair crib door bed stairs window mirror computer
	outside*	sand flowers flower tree trees sun rocks
	places	house room
	people*	baby mommy girl boy babies aunt people sam guy toby
	games & routines*	game nap breaky
verb	transitive verb	painting
	ditransitive verb*	put give putting
	intransitive verb	N/A
	(in)transitive verb*	want see get know look like think try play read got end start
	special verb*	's is do are can have 're s done be did 'm 'll will wanna need

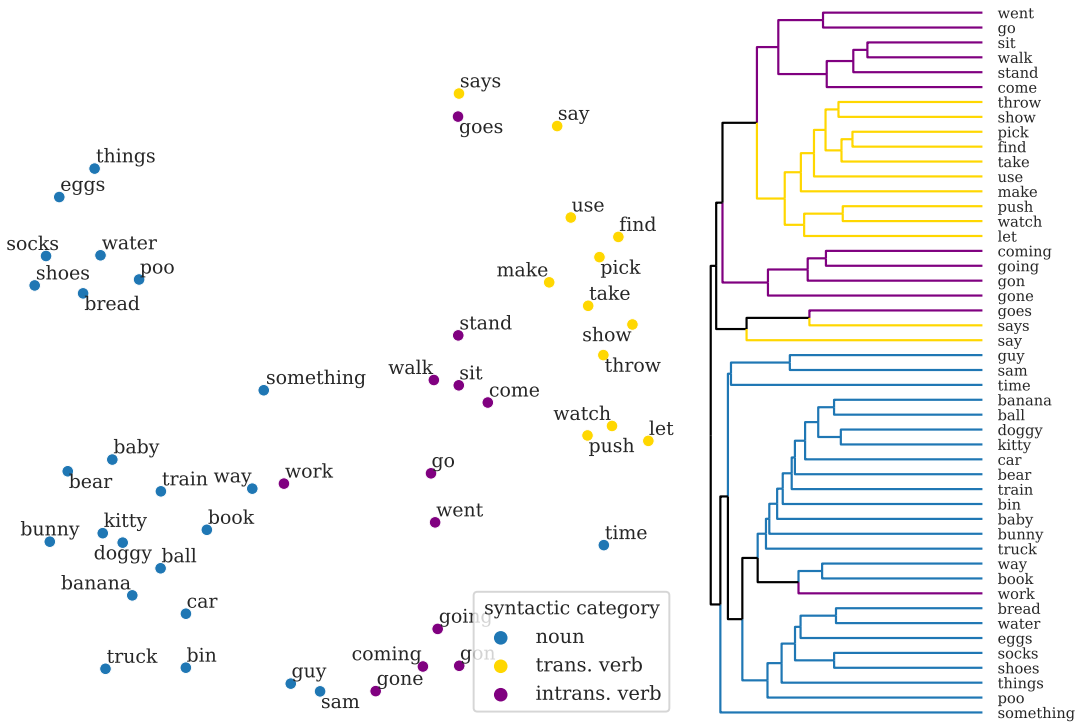
Table 4: Sample words we excluded, arranged into categories we considered they are closest to (for included categories) or they are in (excluded categories are marked with *). N/A means no words are excluded from this category. To illustrate our process to determine whether to include or exclude a word, here is an example: for the word "marmite", we searched on the internet and found it is "a British savoury food spread", but examples in the dataset showed that it is the name of one of the family's cats; however, we still excluded this word because it is not a common word for "animals".

A.4 Additional Clustering Figures

In this section, we include plots demonstrating that the learned networks are sensitive to other kinds of syntactic and semantic structures. First, we show additional t-SNE and dendrogram plots for the CBOW network showing that it can also differentiate syntactic categories, including nouns vs. verbs and finer subcategories of verbs like transitive vs. intransitive verbs (Figure 8), and also different semantic categories (Figure 9). Second, we also show additional t-SNE and dendrogram plots for the Captioning LSTM showing that its representations for words do not change much when given the images (Figure 10 and 11). Then, Figures 12, 13 and 14 show t-SNE and dendrogram plots that include words from additional syntactic categories (adjectives and adverbs) for the LSTM, CBOW and Captioning LSTM networks respectively, showing that all networks form clusters that correspond to each kind of syntactic category. Finally, Figure 15 presents a t-SNE plot showing different word embeddings from the LSTM network colored by animacy.

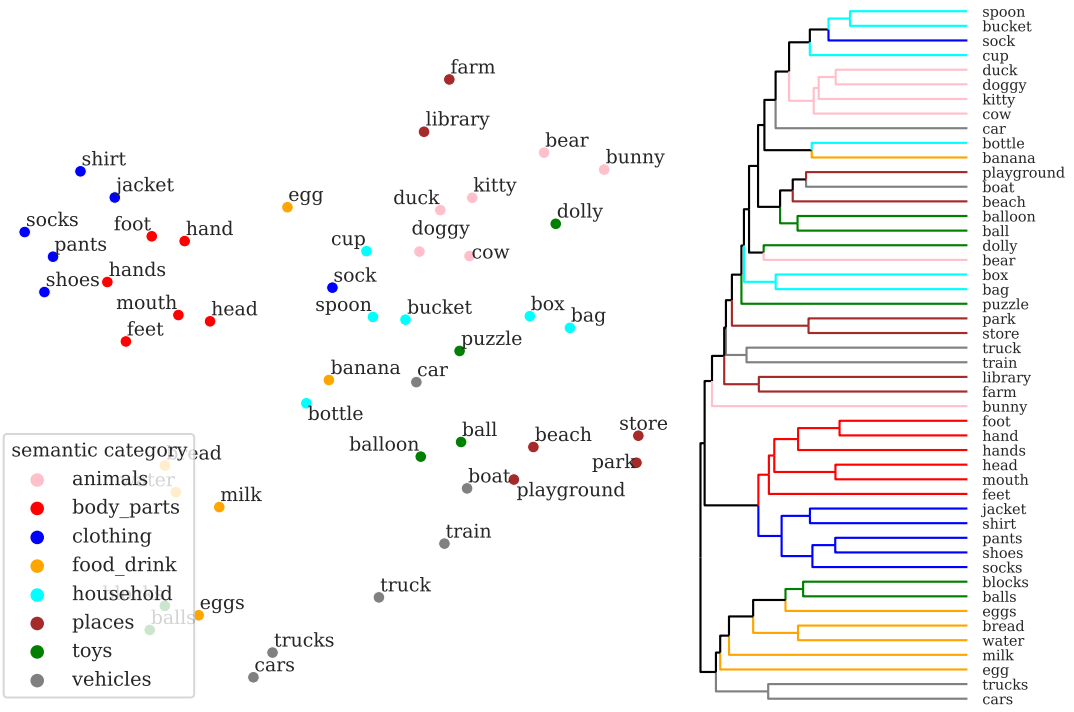
A.5 Cloze Test Details

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. We identify each token that is a noun or verb by using the POS tags labeled by Stanza POS tagger (Qi et al., 2020), and build the vocabulary of word fillers (nouns and verbs to fill into the clozes) by using every word that has its most frequent syntactic category as noun or verb, and is not ambiguous in its syntactic category ($\geq$90% of its occurrences are with its most frequent syntactic category). This resulted in an evaluation set containing 2412 clozes, with 848 (35%) for nouns and 1564 (65%) for verbs, and vocabulary of word fillers containing 1439 words (1040 nouns and 399 verbs). In addition to the language-only models, we also evaluated the



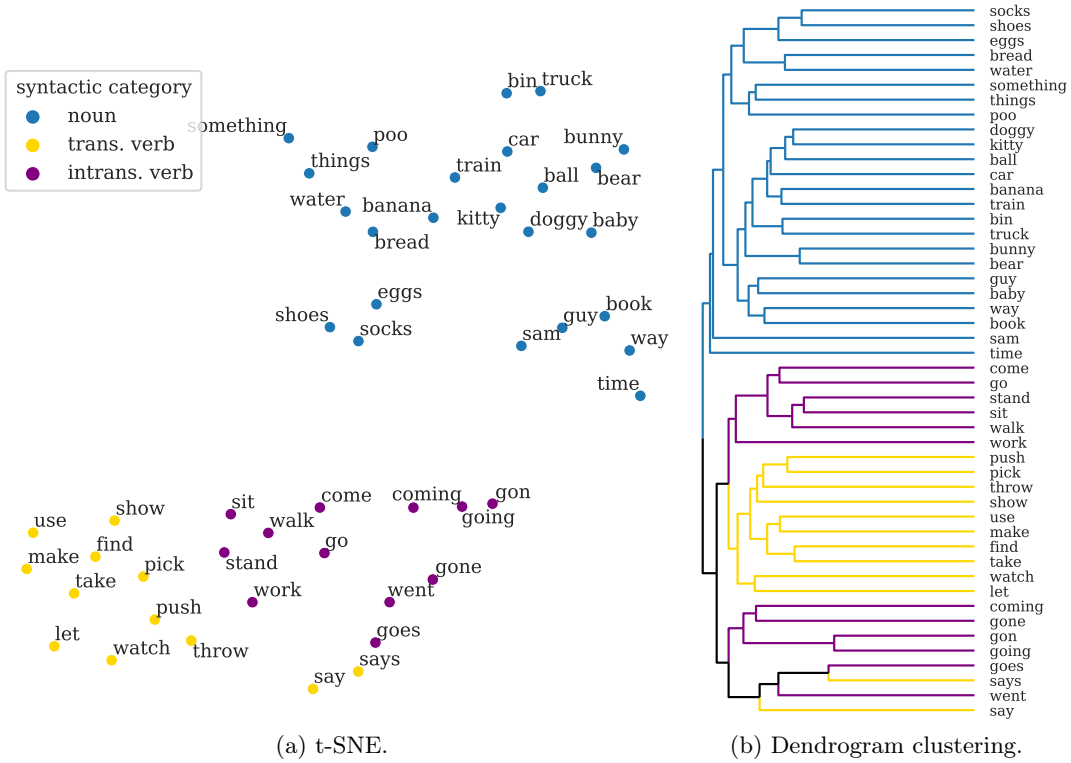
(a) t-SNE. (b) Dendrogram clustering.

Figure 8: Clustering CBOW's word embeddings for syntactic categories by cosine measures in Figure 3. Nouns and verbs form two large clusters. Transitive and intransitive verbs form two smaller subclusters.



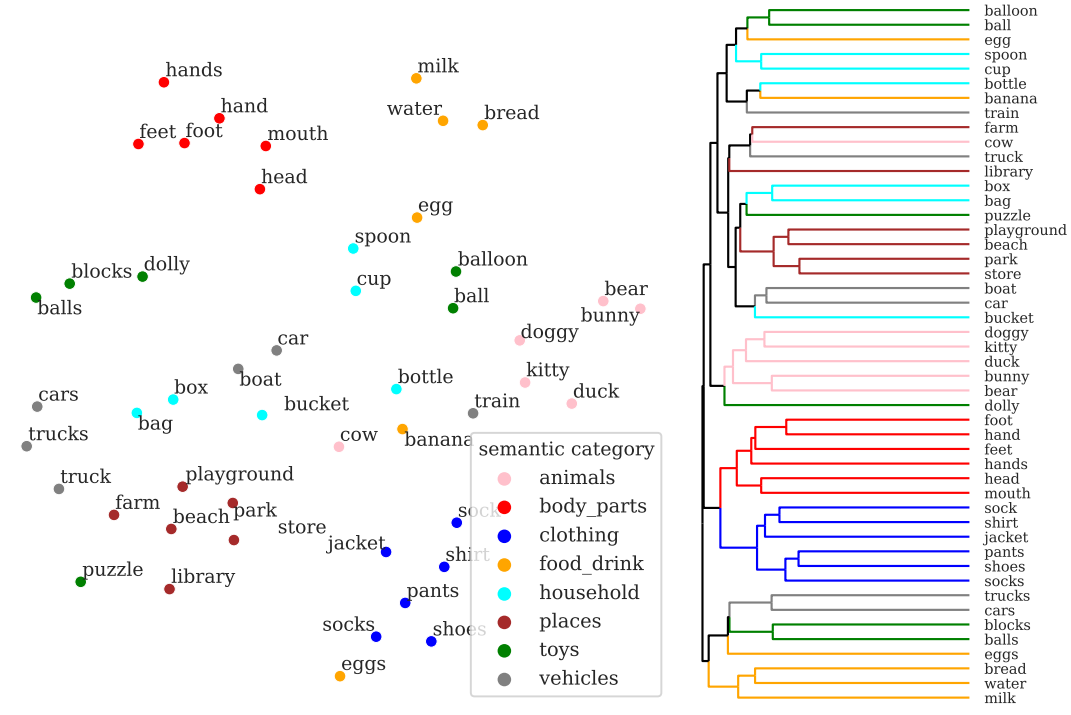
(a) t-SNE. (b) Dendrogram clustering.

Figure 9: Clustering CBOW's word embeddings for semantic categories by cosine measures in Figure 3. The cluster structures are less clear than LSTM's in Figure 15, but several still correspond to semantic categories.



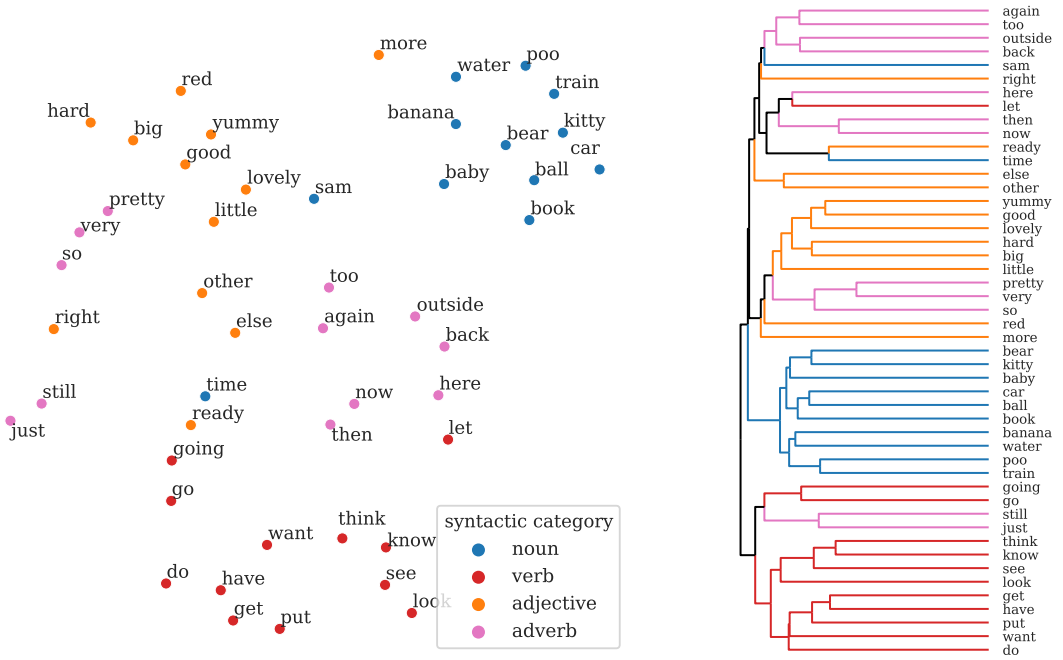
(a) t-SNE. (b) Dendrogram clustering.

Figure 10: Clustering Captioning LSTM's word embeddings for syntactic categories by cosine measures in Figure 3. Compared to Figure 3, the embedding structure is not significantly changed.



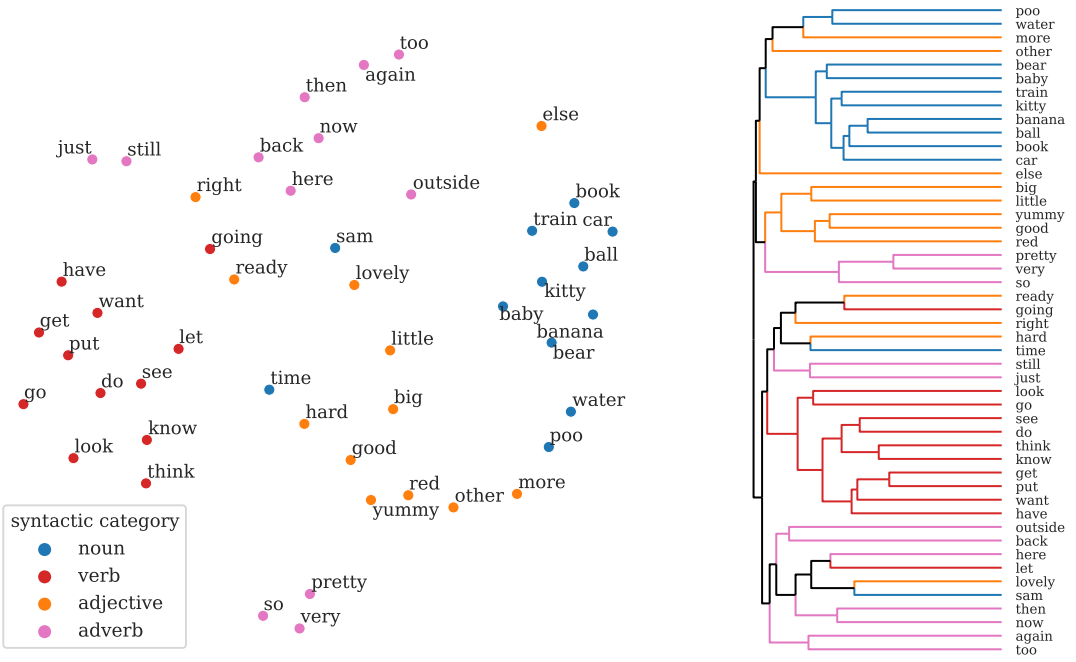
(a) t-SNE. (b) Dendrogram clustering.

Figure 11: Clustering Captioning LSTM's word embeddings for semantic categories by cosine measures in Figure 3. Compared to Figure 4, the embedding structure is not significantly changed.



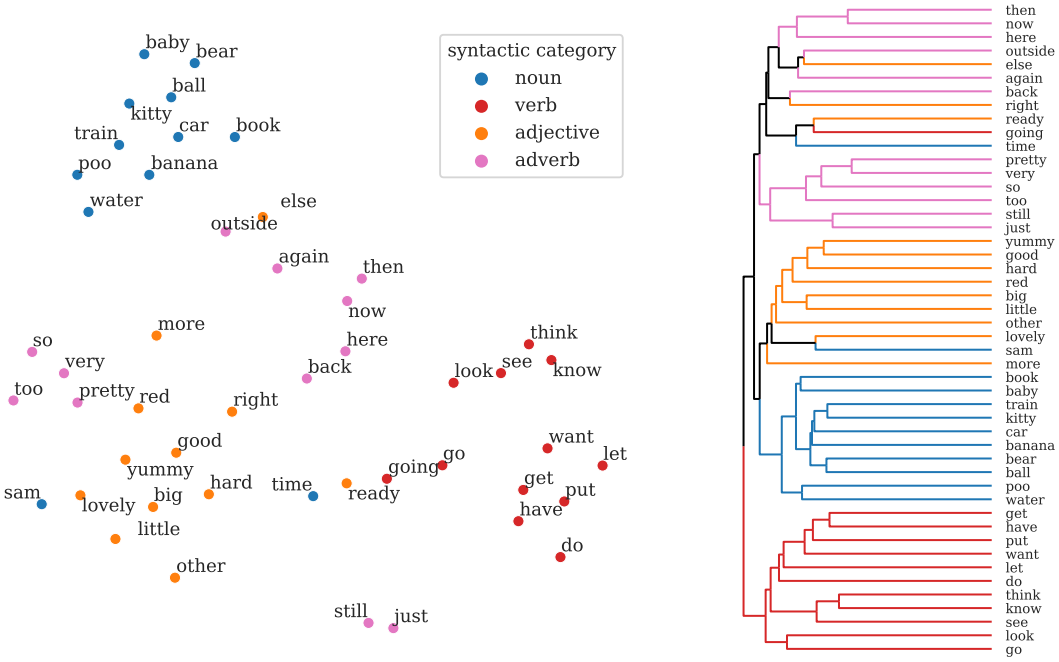
(a) t-SNE. (b) Dendrogram clustering.

Figure 12: Clustering LSTM's word embeddings for more syntactic categories by cosine measures in Figure 3. Clusters generally correspond to syntactic categories.



(a) t-SNE. (b) Dendrogram clustering.

Figure 13: Clustering CBOW's word embeddings for more syntactic categories by cosine measures in Figure 3. The cluster structures are also quite clear compared to LSTM's in Figure 12.



(a) t-SNE. (b) Dendrogram clustering.

Figure 14: Clustering Captioning LSTM's word embeddings for more syntactic categories by cosine measures in Figure 3. The cluster structures are also clear compared to those in Figure 12.

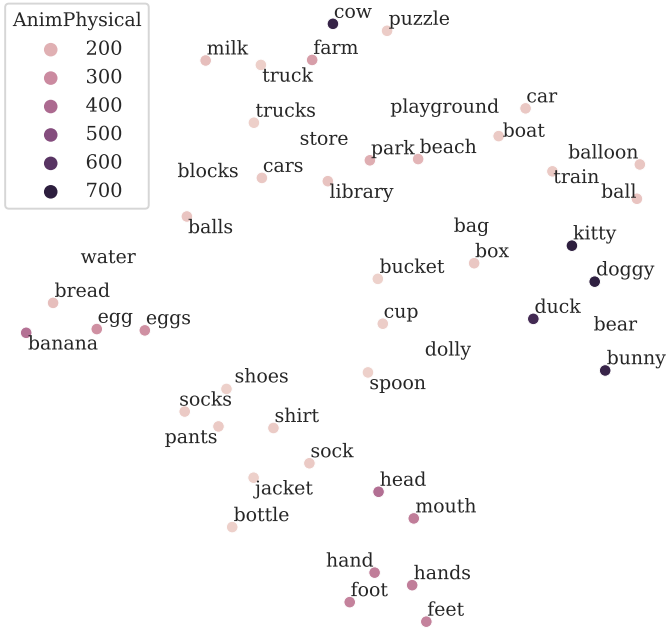


Figure 15: t-SNE of LSTM's word embeddings by distance $1-\cos(u, v)$. The set of words here is the same set in Figure 4. Hue means animacy. Our animacy data is from `https://osf.io/4t3cu/` contributed by Joshua VanArsdall and Janell Blunt. The animacy shown here is from their AnimPhysical field. For 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 left the point with blank color. Comparing this plot to Figure 4, we can see two clusters of animate categories at the bottom-right of the plot, corresponding to body parts and animals. This suggests that the embeddings in this plot may capture some animacy structure, although these results are closely aligned with semantic category structures due to the limited number of words shown in this plot.

Model	Top-5 predictions									
that's an o!										
LSTM	39.7%	egg	8.1%	emu	4.4%	eagle	4.2%	o	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%	o
CBOW	64.1%	hour	19.2%	ant	4.1%	apple	3.8%	emu	2.6%	egg
CBOW	53.3%	hour	16.7%	ant	8.1%	apple	6.6%	emu	5.3%	egg
CBOW	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
CBOW	57.9%	's	20.7%	s	20.5%	is	0.5%	was	0.2%	are
CBOW	57.9%	's	20.8%	s	20.4%	is	0.4%	was	0.2%	are
CBOW	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
CBOW	9.2%	magazine	7.4%	bug	7.3%	moment	7.0%	biscuit	6.5%	horse
CBOW	10.0%	magazine	8.5%	bug	6.9%	moment	6.4%	biscuit	5.9%	horse
CBOW	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
CBOW	61.5%	show	16.9%	give	11.2%	want	6.2%	tell	1.5%	showing
CBOW	62.3%	show	16.6%	give	11.2%	want	5.8%	tell	1.7%	showing
CBOW	63.2%	show	16.3%	give	11.0%	want	5.3%	tell	1.6%	showing
you keep eating.										
LSTM	30.2%	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	2.9%	doing
LSTM	24.2%	going	7.8%	looking	7.1%	doing	5.6%	eating	4.4%	one
CBOW	65.6%	going	14.4%	eating	4.6%	doing	4.6%	holding	2.4%	trying
CBOW	70.8%	going	7.9%	eating	5.6%	holding	3.5%	pressing	2.8%	trying
CBOW	69.2%	going	10.4%	eating	4.5%	trying	3.0%	doing	2.6%	pressing

Table 5: **Additional examples of clozes and the networks' predictions.** Three rows of a same architecture are results from three runs.

Captioning LSTMs by providing them the paired image frames and found they achieve 97.83% ($SD = 0.10\%$ over 3 runs), which is close to the result of language-only LSTMs. 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 and excluding these atypical words. This left 1682 clozes, with 795 (47%) for nouns and 887 (53%) for verbs, and 1406 words in the filling word vocabulary (1034 nouns and 372 verbs). On this filtered set, our language-only LSTMs achieve 97.44% ($SD = 0.10\%$) accuracy, CBOWs achieve 90.35% ($SD = 0.28\%$) accuracy, and Captioning LSTMs achieve 97.42% ($SD = 0.34\%$) 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 5, 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.

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

Table 6: 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.

A.6 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 is 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, 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 6 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 16. The Transformer network performs best on most of the tests. Note also that CBOW and the n-gram models do well on some, but not all subsets, and the LSTM is better overall. The n-gram models serve as baselines indicating whether there are simple, short-distance or word count distributional cues in the data distribution that the model can potentially utilize. This turned out to be true with regard to some of the targeted tests. On the `quantifiers - superlative` subset, the unigram model achieves perfect accuracy because some quantifiers occur more frequently than others in the training data. For example, for superlative quantifiers "at least" and "more than", the product of unigram probabilities given the training corpus is higher for the latter which happens to be always grammatical in this subset. ("at": 682, "least": 11 vs. "more": 504, "than": 39). (Another pair of contrast in this subset, "at most" vs "fewer than", was filtered out because the word "fewer" is out-of-vocabulary.) Another

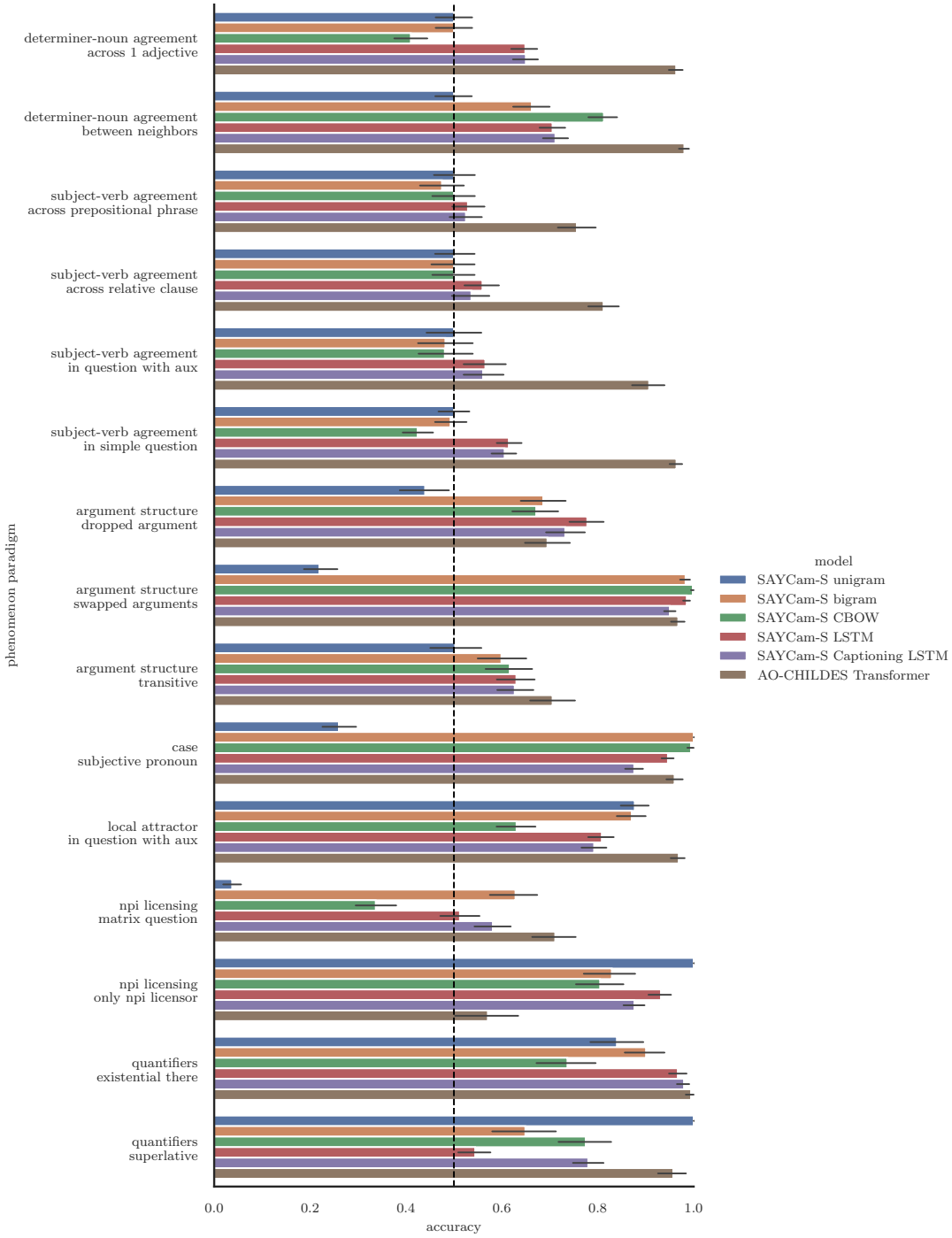


Figure 16: Accuracy on linguistic acceptability tests. The dashed line means the chance level.

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

Table 7: Type-level mean loss difference from language-only LSTM to Captioning LSTM on the validation set, with t-test results. Results on adjective and cardinal number are not significant.

example of short-distance distributional cues: in the `case - subjective pronoun` subset, the nominative case pronoun "I" usually occurs at the beginning of a grammatical sentence, so the bigram model always assigns a lower probability if it occurs not at the beginning of the sentence. The LSTM is better on subsets where longer-distance dependencies are required, such as `determiner-noun agreement - across 1 adjective` and `quantifiers - existential there`. The Captioning LSTM performs mostly close to the language-only LSTM; the only notable difference, shown in Figure 16, is that it is noticeably better on the `quantifiers - superlative` subset, close to the CBOW. We do not have a clear explanation for this performance difference, and more research is needed. Captioning cannot directly help in this task because the synthetic test sentences are not grounded and have no paired images; we simply fed the mean image of the training data (unrelated to the candidate sentences) to the captioning model when testing it on these candidate sentences. Though hypothetically captioning can indirectly help in training a stronger language model, due to the confound of the hidden state initialization in the captioning model and the specific data distribution of this subset.

A.7 Loss Difference between language-only LSTMs and Captioning LSTMs

Table 7 shows statistics of type-level loss difference between language-only LSTM to Captioning LSTM on the validation set, explaining Figure 6.

A.8 Cosine Similarity Heatmaps

Figures 17, 18 and 19 are heatmaps that visualize the cosine similarity matrices between words (corresponding to Figures 3, 10 and 8, respectively) for the LSTM, Captioning LSTM and CBOW models respectively, showing the similarity within and across different syntactic categories. These similarity matrices are used to calculate the Pearson correlations in the Representational Similarity Analysis in Section 4.2.2. Additionally, Figures 20, 21 and 22 are heatmaps that visualize the cosine similarity matrices between another set of words (corresponding to Figures 12, 14 and 13, respectively) for the LSTM, Captioning LSTM and CBOW models respectively, showing the similarity within and across another set of syntactic categories (noun, verb, adjective, adverb), from which we have the same observation as in Section 4.2.2 that LSTM and Captioning LSTM are very similar.

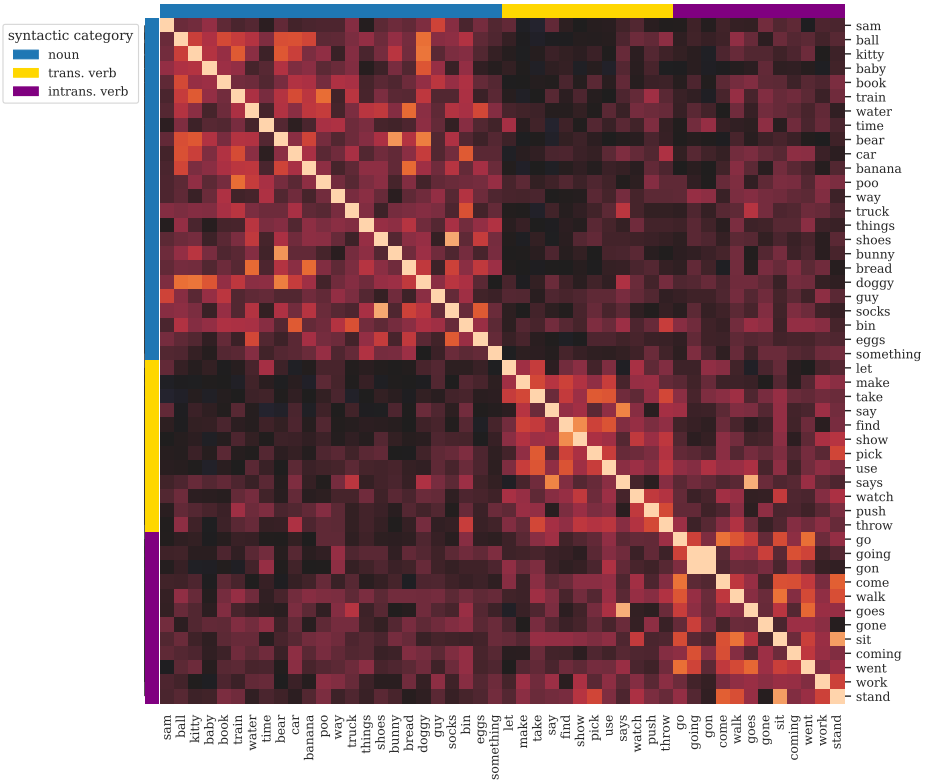


Figure 17: Heatmap of cosine similarity of LSTM's word embeddings. Nouns and verbs are more similar to other words within the same category than other words in the different category.

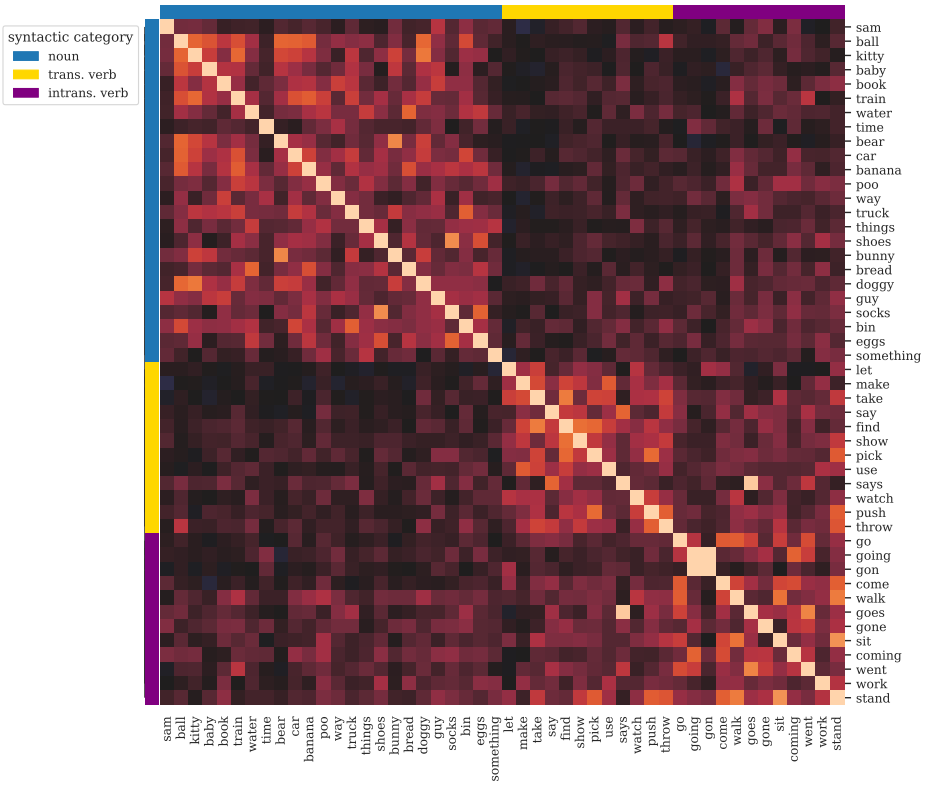


Figure 18: Heatmap of cosine similarity of Captioning LSTM's word embeddings. Nouns and verbs are more similar to other words within the same category than other words in the different category.

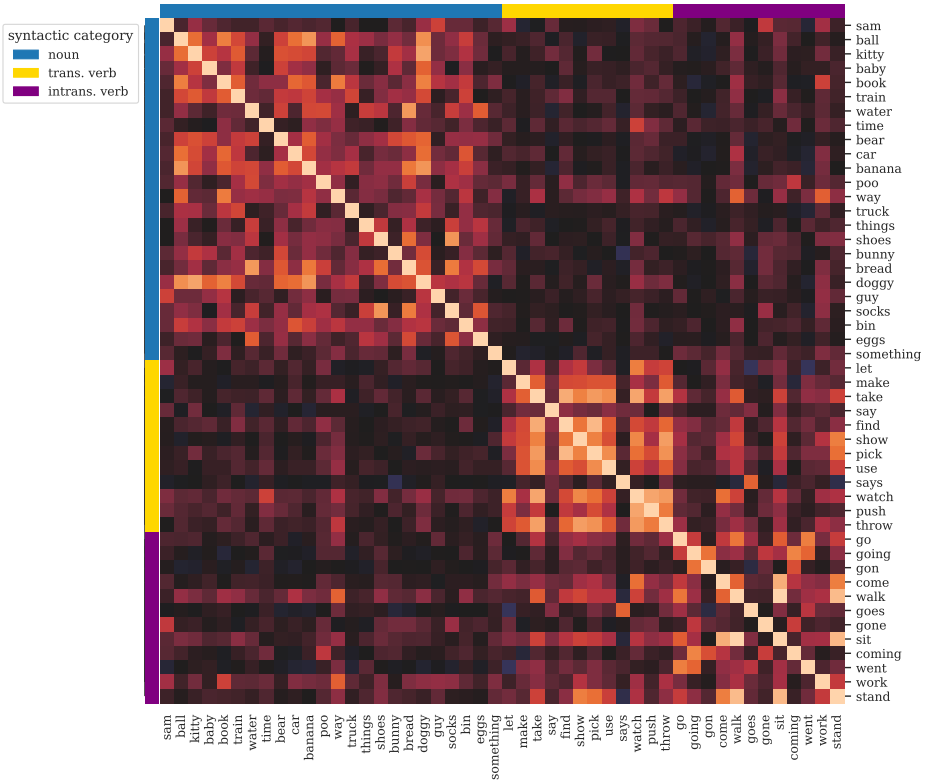


Figure 19: Heatmap of cosine similarity of CBOW's word embeddings. Nouns and verbs are more similar to other words within the same category than other words in the different category.

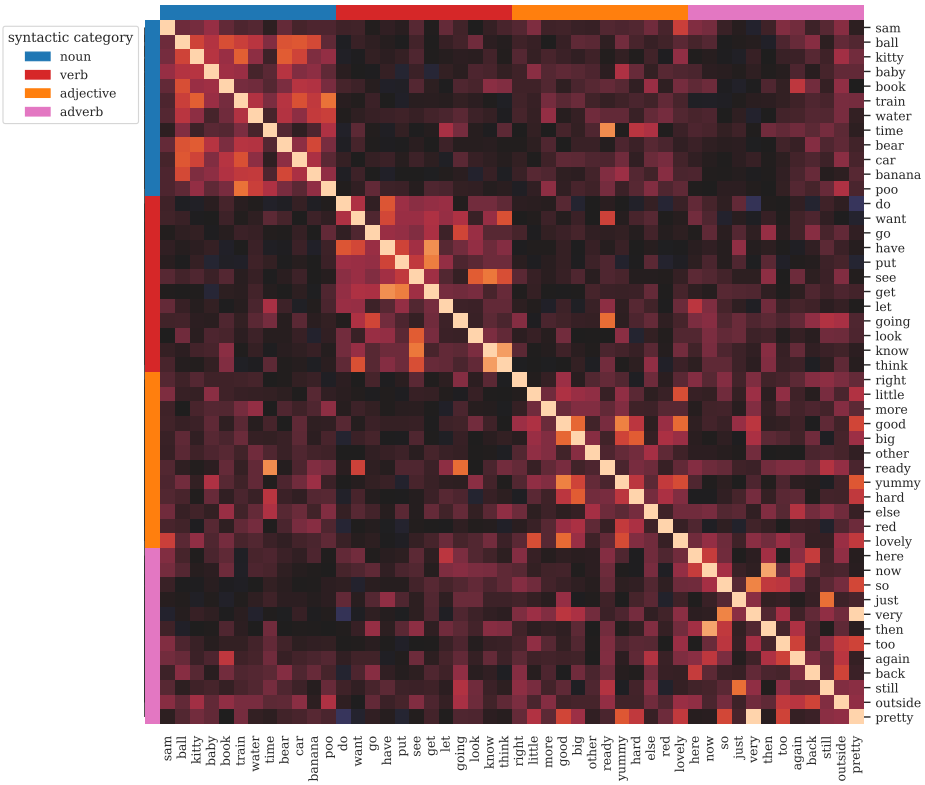


Figure 20: Heatmap of cosine similarity of LSTM's word embeddings. Words are more similar to other words within the same category than other words in the different category.

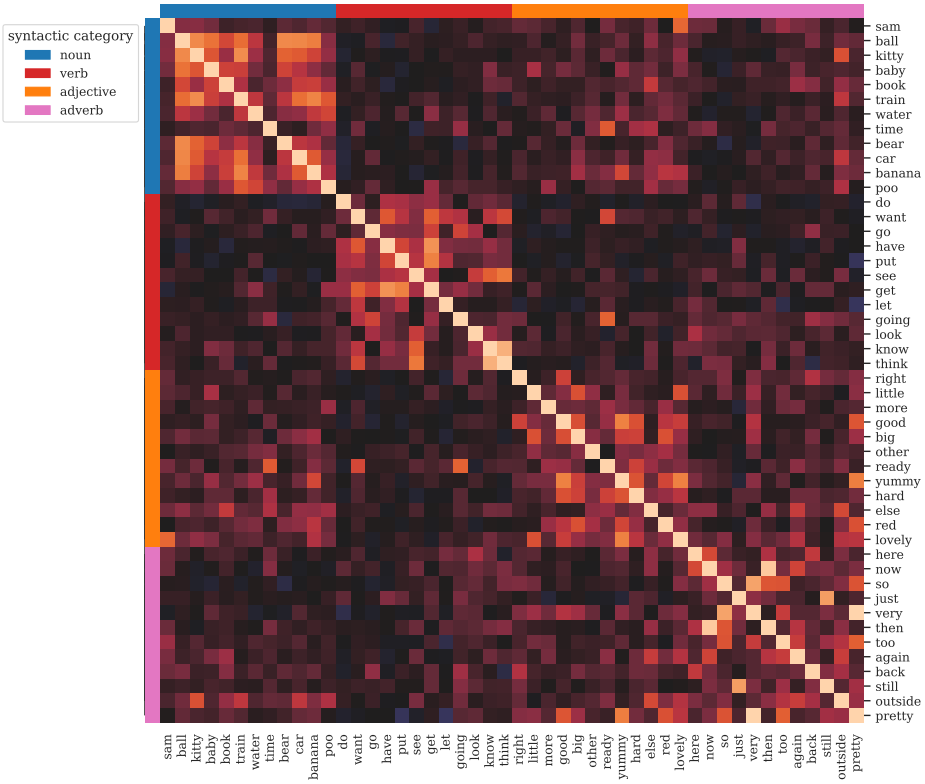


Figure 21: Heatmap of cosine similarity of Captioning LSTM's word embeddings. Words are more similar to other words within the same category than other words in the different category.

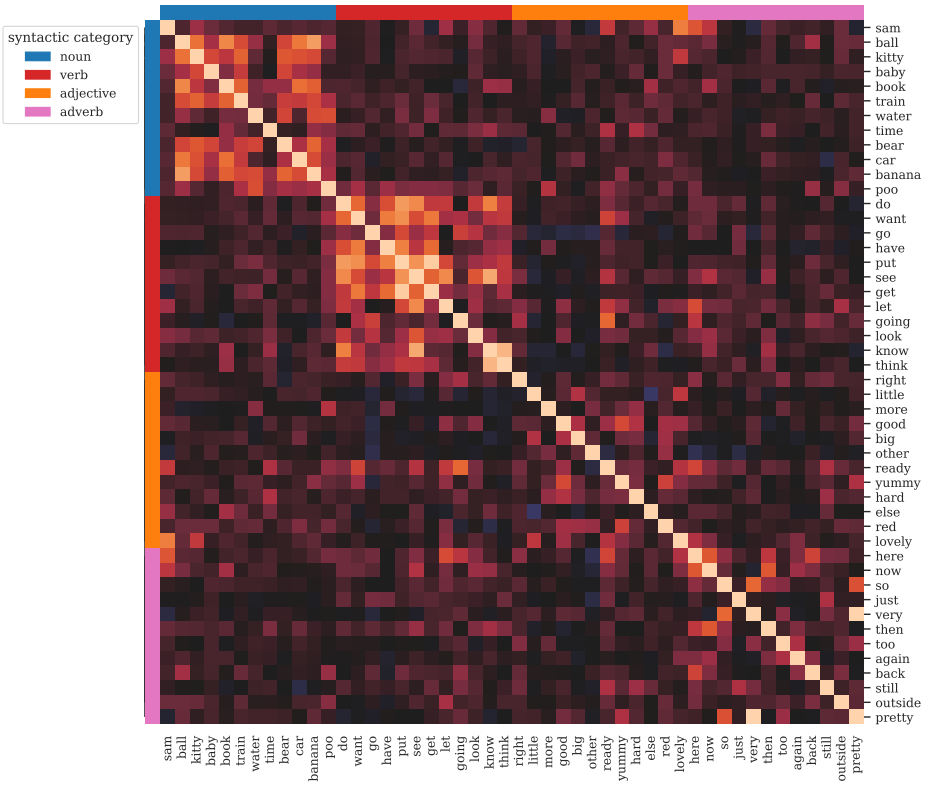


Figure 22: Heatmap of cosine similarity of CBOW's word embeddings. Words are more similar to other words within the same category than other words in the different category.