Poverty of the Stimulus with CHILDES: Supplementary Materials

CHILDES data pre-processing

The first thing I did was that I cleaned up all the transcription marks. I've described what I did in other places, I'll copy paste that here later.

How I split into training, validation, and test sets.

1. Gather the non-child utterances and corresponding filenames.
2. Shuffel by filename.
3. Create a map from file name to number of utterances.
4. Order map by number of utterances.
5. Iterate through sets of n (=30) file names in map, randomly assign one to the validation set, another to the test set, and leave the remainder for the training set.
6. Split data by assignments.

This means that approximatly $\frac{1}{30}$ sentences and $\frac{1}{30}$ files will be in the validation and test sets.

Table 1: Gather the non-child utterances and corresponding filenames

who's a good boy ?	childes/Bates/fred.cha
haha !	childes/Bates/sarah.cha
the doggy ate the bone .	childes/Bates/amy.cha
what did the doggy do ?	childes/Bates/amy.cha

Hyperparameters and further model details

LSTM For LSTMs I explored the following hyperparameters for a total of 48 models:

Table 2: Shuffel by filename

haha !	childes/Bates/sarah.cha
the doggy ate the bone .	childes/Bates/amy.cha
what did the doggy do ?	childes/Bates/amy.cha who's a good boy ?
childes/Bates/fred.cha	

Table 3: Create a map from file name to number of utterances

childes/Bates/fred.cha	1
childes/Bates/sarah.cha	1
childes/Bates/amy.cha	2

Table 4: Order map by number of utterances

childes/Bates/amy.cha	2
childes/Bates/fred.cha	1
childes/Bates/sarah.cha	1

Table 5: Randomly assign to train, valid, and test, in batches

childes/Bates/amy.cha	2	train
childes/Bates/fred.cha	1	valid
childes/Bates/sarah.cha	1	test

Table 6: Split data by assignments

who's a good boy ?	childes/Bates/fred.cha	valid
haha !	childes/Bates/sarah.cha	test
the doggy ate the bone .	childes/Bates/amy.cha	train
what did the doggy do ?	childes/Bates/amy.cha	train

1. layers: 2
2. hidden and embedding size: 200, 800
3. batch size: 20, 80
4. dropout rate: 0.0, 0.2, 0.4, 0.6
5. learning rate: 5.0, 10.0, 20.0