Recurrent babbling: Supplementary Material

Ludovica Pannitto
CIMeC
University of Trento
ludovica.pannitto@unitn.it

Aurélie Herbelot
CIMeC/DISI
University of Trento
aurelie.herbelot@unitn.it

1 Corpus

- For the parser, we used the `english-ewt` model from *Universal Dependencies 2.3 Models*[1]
- Table 1 summarizes some of the features of the three subcorpora we used
- In order to extract portions of 3 million tokens with (almost) uniform probability, we used a slight variation of `reservoir sampling`, the pseudocode listing is provided in Listing 1

2 Language Models

- The bayesian optimization was performed in two steps, the first to isolate the interesting intervals for hyperparameters that would suit all the three subcorpora, and the second one to find the best configuration. Table 2 summarized the ranges used for both steps and Table 3 contains the configurations of the best models for all subcorpora.
- From each model, our aim was to sample approximately the same amount of language provided as input. Since the network was character-based, but the remainder of the extraction process happened on a sentence-by-sentence basis, we had to balance the two aspects. Therefore, we fixed a number of iterations (150), and sampled a starting letter at the beginning of each one, based on the distribution of letters at the beginning of sentences in the original corpus. For each iteration, we sampled a variable number of sentences in order to match the number of sentences in the input (359 for opensubtitles, 159 for simplewikipedia and 597 for childes). We also set a maximum number of characters per sentence (average sentence length of input + 2 standard deviations) as we wanted to avoid loops, expecially in models snapshotted ad earlier epochs.
- The complete set of generated language can be downloaded at https://osf.io/r25at/?view_only=77afc9a31e424f819f4e3e6ea73a2f7f, along with the input data.

3 Catenae

- The recursive algorithm used to extract catenae is reported in Listing 2
- A selection of the most and least associated catenae for each subcorpora can be found in Tables 4, 5 and **??**

4 What do ANNs approximate?

Tables 11, 12, 13, report correlation values (Spearman ρ) computed over top $10K$ catenae extracted from each *babbling* stage, for CHILDES, opensubtitles and simplewikipedia respectively.

Table 14 shows the same values computed among structures extracted from *best models* (BM) and the input corpora.

5 Meaning and abstraction

- Kruskall-Wallis test is significant regardless of the dependent variable (*similarity* or *distributional shift*). When grouping on the distributional shift we considered as negative shift catenae that had average difference below -0.05, as no difference catenae that had average difference between -0.05 and 0.05 and as positive difference catenae that had average difference above 0.05. Results are as follows, as reported in the paper:

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2898, Straka and Straková (2018)

$p = 6.988142426844016e-28$ for cat_1;
$p = 7.420868598608134e-32$ for cat_2.

When grouping on similarity, we considered catenae with average cosine similarity above 0.7 in the *high similarity* group, catenae with average cosine similarity between 0.4 and 0.7 in the *medium similarity* group, and catenae with average cosine similarity below 0.4 in the *low similarity* group. Results are as follows:

$p = 1.070823941185318e-66$ for cat_1;
$p = 1.6648509572822658e-72$ for cat_2

- Posthocs are reported in Tables 7, 8, 9 and 10. Figures 1a, 1b, 1c and 1d show the distributions, respectively.

		CHILDES	opensubtitles	simplewikipedia
	number of sentences	527k	430k	184k
	number of tokens	2,799k	2,421k	2,570k
	vocabulary size	21k	45k	114k
	average sentence length	5.311	5.634	13.965
	average word length	3.403	3.722	4.494
	type-token ratio	0.007	0.018	0.044
	hapax-token ratio	0.003	0.008	0.025
	lexical density	0.543	0.552	0.600
	average depth of sentences	2.779	2.891	4.386
	average *arity* of verbal roots	2.831	4.176	4.948
PoS of root	**Verb**	60.25%	51.18%	59.87%
	Noun	20.90%	22.35%	33.25%
	Adjective	6.13%	9.75%	4.58%
	Pronoun	2.36%	1.16%	0.05%
	Wh-word	3.60%	2.26%	0.08%

Table 1: The table shows some simple figures that highlight some of the differences among the resources we gathered.

	step 1	**step 2**
batch size	20-129	20-40
emsize	40-401	350-500
hidden	40-401	350-500
nlayers	2,4	3,4
dropout	0-0.5	0-0.2
learning rate	0.001-1	0.8-1
epochs	50-150	30-70
seq length	10-101	15-50

Table 2: Hyperparameters' regions used for step 1 and step 2 of the bayesian optimization procedure.

corpus	**target**	**batch**	**emsize**	**hidden**	**nlayers**	**dropout**	**lr**	**epochs**	**bptt**
CHILDES	1.063	28	371	495	3	0.112	0.96	37	39
opensubtitles	1.148	20	353	495	3	0.163	0.92	46	48
simplewikipedia	1.171	28	371	495	3	0.112	0.96	37	39

Table 3: Target values (perplexity) and parameters of the best models selected by the bayesian optimizer.

Listing 1: Reservoir sampling

```
def reservoir_tokens_number(corpus_sentences, number_of_tokens):
  reservoir = []
  len_sampled = []
  considered_tokens = 0
  sentence_number = -1

  for sentence_num, sentence in enumerate(corpus_sentences):
    considered_tokens += len(sentence)
  if considered_tokens < size*1.2:
    reservoir.append(sentence_number)
    len_sampled.append(len(sentence))
```

catena	frequency	mi
largest MI		
@nsubj @root	294.59K	633.93K
@nsubj _VERB	269.81K	621.08K
_DET _NOUN	189.97K	552.32K
@det _NOUN	185.64K	550.92K
_VERB @obj	190.72K	520.82K
_PRON _VERB	271.44K	503.17K
_PRON @root	290.78K	487.42K
@nsubj _AUX @root	129.60K	478.86K
@nsubj _VERB @obj	111.34K	466.75K
_VERB _ADP @obl	68.30K	429.38K
MI near zero		
_NOUN @root @xcomp _NOUN	320	0.04
@cc _PRON @root you	169	0.03
@nsubj _ADV @det _NOUN _VERB	113	-0.01
_PRON want _ADJ	100	0.03
_ADV _AUX _VERB @advmod _NOUN	90	-0.05
_NOUN _NOUN @root _ADJ @obj	79	-0.01
_PRON _VERB @nsubj @aux _ADJ	78	0.03
_VERB _VERB _VERB _VERB @xcomp	77	0
@obl:tmod @aux _VERB	63	-0.03
@nsubj _PART _VERB _PART @root	53	-0.01
smallest MI		
_PRON _PRON	12.74K	-37.53K
_PRON @nsubj	17.50K	-35.54K
@root @nsubj	27.61K	-34.89K
@nsubj _PRON	11.63K	-30.47K
_PRON _AUX	19.00K	-27.03K
_VERB @nsubj	12.79K	-26.82K
_AUX _PRON	15.75K	-26.67K
_VERB @root	8.40K	-25.63K
_NOUN _PRON	8.01K	-24.28K
@root _AUX	21.86K	-24.27K

Table 4: Catenae extracted from CHILDES with their frequency and mutual information.

catena	frequency	mi
largest MI		
@nsubj _AUX @root	115.74K	466.29K
@nsubj _VERB	192.29K	454.95K
@nsubj @root	202.63K	425.80K
_VERB @obj	143.34K	405.82K
_DET _NOUN	146.24K	382.79K
@det _NOUN	142.79K	378.51K
_PRON _AUX @root	107.77K	367.34K
@nsubj @aux _VERB	77.18K	361.89K
_VERB _ADP @obl	60.26K	357.30K
_VERB @case @obl	58.17K	347.78K
MI around zero		
@root @obj _PRON @det _NOUN	200	0.05
i @root _ADP _ADV	156	0
_DET @nsubj _VERB _PRON _ADV	141	0.02
we here	96	0.05
_VERB @amod _NOUN _NOUN _ADV	77	0
@nsubj @root _PRON _PART _ADJ	75	0.04
@advcl @aux @nsubj @root	68	-0.03
_DET @obl you _VERB	62	-0.01
gets _NOUN	61	0
away @obl	59	0.01
smallest MI		
_PRON @nsubj	10.05K	-22.43K
@root @nsubj	9.93K	-22.34K
_PRON _PRON	6.01K	-20.97K
_NOUN _PRON	5.69K	-19.56K
_VERB @root	6.09K	-19.07K
@nsubj _PRON	5.71K	-17.39K
_NOUN @root	48.34K	-16.99K
_PRON _AUX	6.90K	-16.22K
_VERB @nsubj	6.13K	-15.97K
@root _AUX	6.44K	-15.79K

Table 5: Catenae extracted from opensubtitles with their frequency and mutual information.

catena	frequency	mi
largest MI		
_VERB _ADP @obl	76.03K	386.69K
_VERB @case @obl	75.77K	381.07K
@nsubj:pass @aux:pass _VERB _ADP @obl	24.27K	314.61K
@nsubj:pass @aux:pass _VERB @case @obl	24.20K	312.29K
_DET _NOUN	150.28K	311.67K
@det _NOUN	149.04K	309.86K
@nsubj:pass _AUX _VERB _ADP @obl	26.43K	302.78K
@nsubj:pass _AUX _VERB @case @obl	26.35K	300.35K
@nsubj @root	96.50K	299.59K
_DET _NOUN @case @nmod	51.00K	295.75K
MI around zero		
@det _PROPN _ADP _PROPN _NOUN	545	-0.04
_PROPN _PROPN @nsubj _NOUN _VERB	293	0.02
@det _NOUN _PROPN _AUX _PROPN	270	0.02
@det @obl @root @nmod	195	-0.03
_ADP _NUM _NOUN _ADP _PROPN	194	0.03
_PROPN is @compound _PROPN	178	0.02
@nsubj @det _NOUN @case @obl	145	0.03
@root _DET _NOUN @case _PRON	142	0.04
_AUX @root _ADP _ADP _PROPN	130	-0.04
_DET _PROPN @compound @nmod _NOUN	127	-0.01 **smallest MI**
_PROPN _NOUN	34.51K	-13.55K
_PROPN @case	4.77K	-12.51K
_NOUN @nsubj	6.02K	-12.44K
_NOUN _ADJ	8.04K	-12.13K
_NOUN @obl	8.49K	-11.03K
_NOUN @case	2.32K	-9.78K
@nmod _NOUN	8.38K	-9.70K
@case @root	4.27K	-9.25K
_ADP @root	4.26K	-9.02K
_NOUN @compound	3.02K	-8.89K

Table 6: Catenae extracted from simplewikipedia with their frequency and mutual information.

	negative	none	positive
negative	-	7.32e-01	1.33e-03
none	0.732	-	5.71e-29
positive	0.001	5.71e-29	-

Table 7: CAT1 posthoc diffs

	negative	none	positive
negative	-	6.83e-06	4.57e-05
none	0.000	-	4.15e-29
positive	0.000	4.15e-29	-

Table 8: CAT1 posthoc diffs

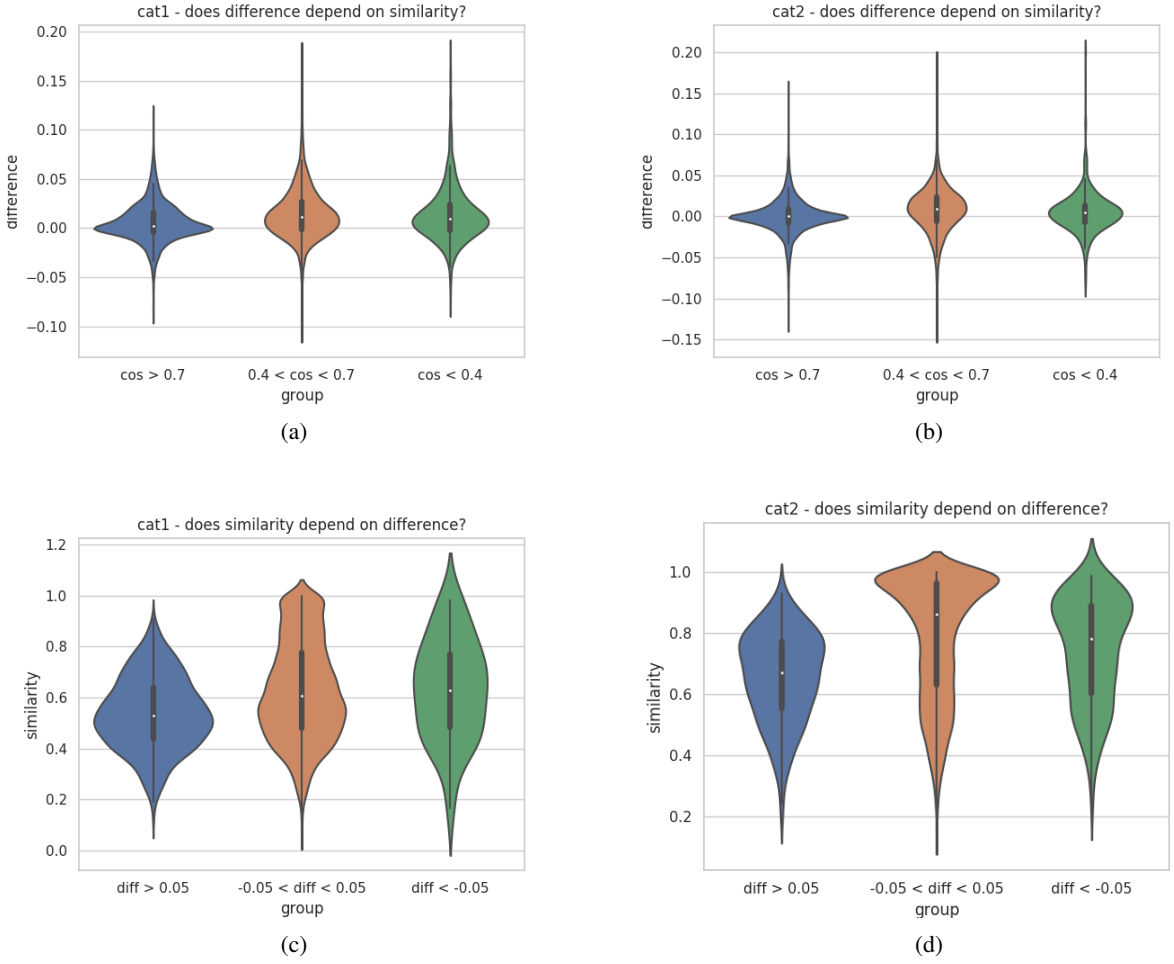

Figure 1

	$\cos < 0.4$	$0.4 < \cos < 0.7$	$\cos > 0.7$
$\cos < 0.4$	-	1.297866e-02	1.531517e-19
$0.4 < \cos < 0.7$	1.297866e-02	-	5.071673e-67
$\cos > 0.7$	1.531517e-19	5.071673e-67	-

Table 9: CAT1 cos

	$\cos < 0.4$	$0.4 < \cos < 0.7$	$\cos > 0.7$
$\cos < 0.4$	-	2.296763e-04	1.829284e-05
$0.4 < \cos < 0.7$	0.000230	-	4.158696e-73
$\cos > 0.7$	0.000018	4.158696e-73	-

Table 10: CAT2 cos

```
  else:
    j = random.randrange(sentence_num)
    if j < len(reservoir):
      reservoir[j] = sentence_num
      len_sampled[j] = len(sentence)

  x = 0
  i = 0
  while x < size and i < len(reservoir):
    x += len_sampled[i]
  i = i+1
  return list(sorted(reservoir[:i]))
```

Listing 2: Recursive algorithm for the extraction of catenae

```
def recursive_C(A, tree_children, th=5):
  # if A is a leaf
  if A not in tree_children:
    return [[A]], [[A]]
  else:
    found_catenae = []
    list_of_indep_catenae = [[[A]]]
    for a_child in tree_children[A]:
      c, all_c = recursive_C(a_child, tree_children)
      found_catenae += all_c
      list_of_indep_catenae.append([[None]] + c)

    X = []
    for tup in itertools.product(*list_of_indep_catenae):
      new_catena = list(sorted(filter(
          lambda x: x is not None, sum(tup, []))))
      if len(new_catena) <= th:
        X.append(new_catena)

  return X, X+found_catenae

def extract(sentence):
  children = {}
  tokens = {}
  postags = {}
  rels = {}
  excluded_relations = ["discourse", "fixed", "flat", "comound",
                        "list", "parataxis", "orphan", "goeswith",
                        "reparandum", "punct", "dep"]
  for token in sentence:
    position, word, lemma, pos, _, morph, head, rel, _, _ = token
    if not pos == "PUNCT" and not rel in excluded_relations:
      if head not in children:
        children[head] = []
      children[head].append(position)
      tokens[position] = word
      postags[position] = "_"+pos
```

```
        rels[position] = "@"+rel

    if 0 in children:
      root = children[0][0]
      _, catenae = recursive_C(root, children)

    for catena in catenae:

      tokensandpostags = [[tokens[x] for x in catena],
                          [postags[x] for x in catena],
                          [rels[x] for x in catena]]

      temp = [(0, 1, 2)] * len(catena)
      X = list(itertools.product(*temp))

      for c in X:
        cat = []
        for i, el in enumerate(c):
          cat.append(tokensandpostags[el][i])
        cat = tuple(cat)
        if len(cat) > 1:
          yield cat
```

	CHILDES								
	EPO5	EPO10	EPO15	EPO20	EPO25	EPO30	EPO35	BM	input
EPO5	1	0,922	0,925	0,892	0,894	0,882	0,924	0,914	0,897
EPO10	0,919	1	0,93	0,935	0,924	0,917	0,946	0,951	0,924
EPO15	0,912	0,923	1	0,925	0,929	0,92	0,955	0,949	0,934
EPO20	0,887	0,929	0,916	1	0,894	0,885	0,94	0,942	0,921
EPO25	0,877	0,912	0,919	0,902	1	0,941	0,947	0,945	0,923
EPO30	0,87	0,912	0,92	0,899	0,938	1	0,94	0,946	0,936
EPO35	0,921	0,944	0,952	0,945	0,95	0,942	1	0,983	0,961
BM	0,906	0,945	0,945	0,946	0,949	0,946	0,983	1	0,96
input	0,89	0,908	0,931	0,911	0,913	0,936	0,954	0,953	1

Table 11

	OPENSUBTITLES										
	EPO5	EPO10	EPO15	EPO20	EPO25	EPO30	EPO35	EPO40	EPO45	BM	input
EPO5	1	0,851	0,883	0,907	0,879	0,892	0,911	0,913	0,907	0,915	0,87
EPO10	0,875	1	0,91	0,897	0,912	0,908	0,921	0,928	0,927	0,932	0,896
EPO15	0,871	0,894	1	0,904	0,918	0,929	0,933	0,938	0,911	0,94	0,894
EPO20	0,916	0,882	0,918	1	0,942	0,924	0,944	0,958	0,94	0,954	0,931
EPO25	0,903	0,906	0,934	0,951	1	0,938	0,951	0,966	0,954	0,961	0,942
EPO30	0,888	0,905	0,942	0,914	0,929	1	0,941	0,953	0,916	0,946	0,918
EPO35	0,881	0,832	0,906	0,896	0,873	0,906	1	0,935	0,946	0,948	0,884
EPO40	0,914	0,905	0,943	0,957	0,952	0,958	0,973	1	0,966	0,99	0,952
EPO45	0,905	0,829	0,911	0,906	0,874	0,908	0,978	0,953	1	0,978	0,891
BM	0,915	0,89	0,944	0,944	0,931	0,948	0,983	0,988	0,98	1	0,937
input	0,877	0,892	0,895	0,936	0,94	0,924	0,927	0,956	0,928	0,949	1

Table 12

	SIMPLEWIKIPEDIA								
	EPO5	EPO10	EPO15	EPO20	EPO25	EPO30	EPO35	BM	input
EPO5	1	0,882	0,897	0,893	0,875	0,879	0,894	0,887	0,843
EPO10	0,878	1	0,919	0,909	0,909	0,909	0,952	0,908	0,884
EPO15	0,899	0,918	1	0,908	0,916	0,927	0,954	0,956	0,912
EPO20	0,856	0,914	0,91	1	0,928	0,93	0,937	0,913	0,875
EPO25	0,822	0,905	0,908	0,927	1	0,967	0,925	0,928	0,871
EPO30	0,815	0,897	0,907	0,91	0,961	1	0,919	0,921	0,873
EPO35	0,896	0,948	0,955	0,934	0,929	0,937	1	0,946	0,923
BM	0,878	0,9	0,957	0,91	0,926	0,917	0,942	1	0,911
input	0,856	0,872	0,901	0,868	0,87	0,874	0,914	0,900	1

Table 13

		CHILDES		OPENSUBTITLES		SIMPLEWIKIPEDIA	
		BM	input	BM	input	BM	input
CHILDES	BM	1	0,96	0,632	0,62	0,283	0,276
	input	0,953	1	0,646	0,636	0,285	0,275
OPENSUBTITLES	BM	0,712	0,708	1	0,937	0,315	0,311
	input	0,729	0,739	0,949	1	0,347	0,344
SIMPLEWIKIPEDIA	BM	0,317	0,307	0,346	0,342	1	0,911
	input	0,319	0,304	0,343	0,342	0,900	1

Table 14

References

Milan Straka and Jana Straková. 2018. Universal dependencies 2.3 models for UDPipe (2018-11-15). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.