Recurrent Networks are (Linguistically) Better? An Experiment on Small-LM Training on Child-Directed Speech in Italian

Achille Fusco[1,†], Matilde Barbini[1,†], Maria Letizia Piccini Bianchessi[1,†], Veronica Bressan[1,†], Sofia Neri[1,†], Sarah Rossi[1,†], Tommaso Sgrizzi[1,†], Cristiano Chesi[1,*,†]

[1] *NeTS Lab, IUSS Pavia, P.zza Vittoria 15 27100 Pavia, Italy*

Abstract

Here we discuss strategies and results of a small-sized training program based on Italian child-directed speech (less than 3M tokens) for various network architectures. The rationale behind these experiments [1] lies in the attempt to understand the effect of this naturalistic training diet on different models' architecture. Preliminary findings lead us to conclude that: (i) different tokenization strategies produce mildly significant improvements overall, although segmentation aligns more closely with linguistic intuitions in some cases, but not in others; (ii) modified LSTM networks (eMG-RNN variant) with a single layer and a structurally more controlled cell state perform slightly worse in training loss (compared to standard one- and two-layered LSTM models) but better on linguistically critical contrasts. This suggests that standard loss/accuracy metrics in autoregressive training procedures are linguistically irrelevant and, more generally, misleading since the best-trained models produce poorer linguistic predictions ([2], pace [3]). Overall, the performance of these models remains significantly lower compared to that of 7-year-old native-speaker children in the relevant linguistic contrasts we considered [4].

Keywords

LSTM, Transformers, Small Language Models (SLM), tokenization, cell state control, LM evaluation

1. Introduction

According to the mainstream LLM development pipeline, Transformer-based architectures [5] outperform sequential training models, like LSTM [6], in various NLP tasks. When small-sized training data are available, optimization becomes necessary [7], [8], but common optimization techniques neglect the linguistically relevant fact that these models (i) conflate semantic/world knowledge with morpho-syntactic competence, (ii) require unreasonable training data compared to that needed by children during language acquisition, (iii) the higher their performance, the lower their return in cognitive/linguistic terms [9]. In this paper we address these three issues, starting from the observation that while world knowledge uses all training data available, and the more the better, structural (morpho-syntactic and compositional semantic) knowledge might require a much smaller dataset (from 10 to 100 million words, according to [10]). We explore this intuition further and, based on prolific literature from the '80s showing that typical child errors are structurally sensitive and never random [11], we model networks' architecture to bias learning towards plausible structural configurations, possibly preventing these "small" language models (SLM) from producing wrong linguistic generalizations. We started from a mild revision of the LM training and evaluation pipeline for Italian including alternative approaches to tokenization based on pseudo-morphological decomposition (§2.2); we then approached a more structurally-driven update

CLiC-it 2024: Tenth Italian Conference on Computational Linguistics, Dec 04 — 06, 2024, Pisa, Italy
*Corresponding author.
†These authors contributed equally.
cristiano.chesi@iusspavia.it (C. Chesi)

0000-0002-5389-8884 (A. Fusco); 0009-0007-7986-2365 (M. Barbini); 0009-0005-8116-3358 (M. L. Piccini Bianchessi); 0000-0003-3072-7967 (V. Bressan); 0009-0003-5456-0556 (S. Neri); 0009-0007-2525-2457 (S. Rossi); 0000-0003-1375-1359 (T. Sgrizzi); 0000-0003-1935-1348 (C. Chesi);

of the cell state in LSTM networks, which we will call eMG-RNN variants (§2.3); we finally adopted a precise testing benchmark for specific linguistic contrasts in Italian following BLiMP design [12] (§2.4). We will first set the stage in section (§2) and discuss one alternative tokenization strategy (MorPiece). A simple modification to the gating system in LSTM is proposed that mimics certain linguistic constraints. Then, we will describe the relevant experiments we have run (§3) and draw some conclusions based on the observed results (§4). A general discussion with a description of the next steps will conclude this paper (§5).

2. Revisiting LM training pipeline

LM training pipeline is relatively rigid: after corpus cleaning (i), the data are prepared/optimized for tokenization (ii), then the tokenized input is batched for training autoregressive models (iii), mostly feeding transformer-based architectures (iv). Once the models are trained, the evaluation step requires their assessment using some standard tasks (v). In the next sub-sections, we will identify various criticalities in this pipeline, eventually proposing strategies to mitigate these problems and, in the end, training linguistically more informative SLM.

2.1. Corpus creation and cleaning

The primary data we collected for Italian replicates plausible linguistic input that children may be exposed to during acquisition, in line with [1]. It consists of about 3M tokens divided into child-directed speech (CHILDES Italian section), child movie subtitles (from OpenSubtitles), child songs (from Zecchino D'Oro repository), telephone conversations (VoLIP corpus, [13]), and fairy tales (all from copyright expired sources). Simple cleaning consisted of removing children's productions from CHILDES files as well as any other metalinguistic annotation (speakers' identification, headers, time stamps, tags, links, etc.). Dimension and rough lexical richness of each section are reported in Table 1 (Type-Token Ratio, TTR) before and after the cleaning procedure.

Table 1
Corpus profiling before (bc) and after (ac) cleaning.

Section	tokens bc	tokens ac	TTR
Childes	405892	346155	0.03
Subtitles	959026	700729	0.05
Conversations	80826	58039	0.11
Songs	240309	222572	0.08
Fairy tales	1103543	1287826	0.05
Total	2973879	2431038	0.03

2.2. Tokenization: MorPiece (MoP)

Popular vLLMs use either Byte-Pair Encoding (BPE) [14], [15] or (fast)WordPiece (fWP) [16] algorithms for tokenization. The simplicity and computational efficiency of these approaches contrast with the limited morphological analysis they provide. In rich inflectional languages (e.g., Italian) and agglutinative languages (e.g., Finnish), this might induce linguistically unsound generalizations. Here, we explore a more morphologically informed strategy, inspired by the Tolerance Principle (TP) and Sufficiency Principle (SP) [17], aiming to break words into potentially relevant morphemes without relying on morpheme tables [18]. The experiments we conduct compare the impact of different strategies when integrated into various network architectures. We refer to *MorPiece* (*MoP*) as a TP/SP-based strategy, which can be algorithmically described as follows: each token is traversed from left to right to create a "root trie," and from right to left to create an "inflectional trie" [19]. Each time a node N of the trie is traversed (corresponding to the current character path in the word), the frequency counter associated with this node (N_c) is updated (+1). Nodes corresponding to token endings (characters before white spaces or punctuation) are flagged. Once both tries are created, the optimization procedure explores each descendant, and for every daughter node D_k its frequency k is compared to H_N, the approximation of the harmonic number for N used both in TP and SP [17], where c is the frequency of the mother node N_c:

$$H_N = c/ln(c) \quad \text{(F1)}$$

If $k > H_N$ and $c \neq k$, a productive boundary break is postulated (based on the inference that since there are different continuations and some of them are productive, i.e. sufficiently frequent according to SP, those might be real independent morphemes). We can check if this break respects H_D for the relevant nodes D_j and N_i in the "inflectional trie". This means there exists a path where the frequency i of the daughter node N_i (in the "inflectional trie" the dependency between D and N is reversed) is lower than $j/ln(j)$, where j is the frequency of the mother node D_j. If this is the case, the continuation is not considered "an exception", in the sense of TP [17], suggesting that the continuation is, in fact, a productive independent morpheme. A "++" root node is then activated, the node D_k linked to it, and so on recursively, following the FastWordPiece tokenization strategy [20]. During recognition, the LinMaxMatch identification approach is adopted, as in FastWordPiece. Figure 1 illustrates the relevant morpheme breaks (indicated as "||") obtained by applying this morpheme-breaking procedure in the *root* and *infl* tries fragments.

Various parametric controls have been considered to tune this procedure: (i) a *branching factor* (*bf*) parameter that excludes nodes with an excessively high number (> *bf*) of continuations (the rationale being that when too many continuations are present, they are unlikely to correspond to inflections; this often happens near the root of each trie); (ii) a *cutoff* parameter indicating the lower frequency boundary for a mother node (this is necessary to ensure a minimum number of observations; for example, if *cutoff* = 8, we exclude from the "root" trie any branching daughter with a frequency < 5). As in BPE, minimum frequency control for tokens is also implemented to exclude infrequent dictionary entries.

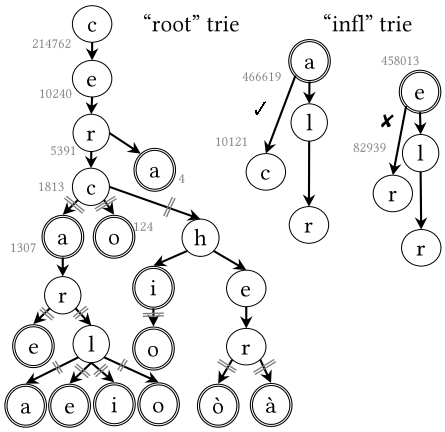

Figure 1: Visualization of a fragment of the "root" and the "infl(ectional)" trie created by MorPiece on our corpus (*cutoff*=100, *bf*=10).

Consider the word "cerca" ("to search for") represented in the "root" trie. In the last "c-a" the relation between H_{fc} and "a" frequency indicates that a break might exist between the nodes "c" (frequency=1813) and "a" (frequency=1307), since H_{fc} = 1813/ln(1813) and 1307 > H_{fc}. This hypothesis is confirmed by the failure of the H_{fc} check at the relevant "infl" "a-c" segment ("a" frequency=10121, "c" frequency=466619): 10121 < 466619/ln(466619). If H_{fc} had been greater than "a" frequency, then no segmentation advantage would have been observable.

The proposed algorithm has a linear time complexity of $O(2n)$, as each trie must be explored deterministically exactly once to evaluate the $H_{N/D}$ frequency relation. The best linguistic results (relatively linguistically coherent segmentations) for our Italian corpus were obtained with *cutoff*=100 and *bf*=10. We found that it was unnecessary to filter the proposed inflectional breaks using the *infl* trie double check (TP) since the LinMaxMatch strategy already efficiently filtered out initially overestimated breaks. However, as an anonymous reviewer correctly pointed out, this strategy does not guarantee total inclusion of every token of our training corpus (in contrast to BPE, for instance). We acknowledge this limitation, but we emphasize that our goal was to produce a smaller, potentially more efficient lexicon. In our experiments, while BPE generated a lexicon of 96028 tokens (67169 when the minimum lexical frequency was set to 2), MoP produced a lexicon of just 55049 tokens (*cutoff*=100, *bf*=10).

2.3. Revisiting LSTM architecture

Despite many variants of the standard LSTM architectures, notably Gated Recurrent Units [21] or LSTM augmented with peephole connections [22], and the discouraging equivalence results for these variations [23], we observe a recent revival of RNN-based model architectures [24]. We believe, in fact, that the core intuition behind the LSTM architecture may be linguistically relevant and worth exploring further, although generally more performant models (for instance in terms of GLUE benchmark, [25]) are usually preferred [26]. The linguistic intuition is that the "long-term memory" (cell state *C* in Figure 2) in LSTM networks could effectively model various types of non-local dependencies using a single mechanism. Linguistically speaking, filler-gap dependencies (1) and co-referential dependencies (2) are both "non-local dependencies" but they are subject to non-identical locality conditions:

(1) a. cosa$_i$ credi che abbia riposto _$_i$?
what (you) believe that (he) shelved?
what do you believe he shelved?
b. *cosa$_i$ credi che abbia riposto il libro [$_{AdvP}$ senza leggere _$_i$]]?
b'. cosa$_i$ credi che abbia riposto _$_i$ [$_{AdvP}$ senza leggere _$_i$]]?
*what do you believe he shelved (*the book) without reading?*

(2) a. [il panino]$_i$, chi credi che lo$_i$ abbia mangiato?
the sandwich, who (you) believe it has eaten?
b. *[il panino]$_i$, chi credi che _$_i$ abbia mangiato?
the sandwich, who (you) believe has eaten?
*the sandwich, who do you believe have eaten *(it)?*

While both dependencies require C(onstituent)-command generalizations to be captured [27], the *adjunct island* in (1), [28], but not *clitic left-dislocation* in (2), [29], can, for instance, be licensed with a(n extra) gap (1).b'. Aware of these differences, we decided to simply alter the gating system to allow the LSTM to create distinct pathways: one to "merge" new tokens, the other to decide if a long-distance dependency is necessary, and subsequently to "move" the relevant items [30]. The processing implementation of these operations is

inspired by expectation-based Minimalist Grammars formalism, eMG [31], and it is then named eMG-RNN.

Following this implementation, *merge* applies incrementally, token by token, and *move* means "retain in memory". In more detail, the cell of an eMG-RNN network performs the forward processing described in the computational graph in Figure 2: (i) the input at time t (x_t) is linearly transformed to a lower dimension vector (E, loosely used for "embedding"), then concatenated (C) with the previous hidden state/output, if any (h_{t-1}). Two pathways, both transformed using a sigmoid function (σ), lead, on the one hand, to the *move* gate, on the other, to the *merge* gate. In the first case, the result of the sigmoid transformation is multiplied ($\odot$, the Hadamard product) with the input (this either erases or allows some component of the original vector to be added (+) to the previous (if any) context/cell state (c_{t-1}) as in LSTM *forget* gate). The *merge* gate, on the other direction, will privilege the new token if the result of the sigmoid combination of the incoming token and the previous hidden state is low, otherwise (1 - this activation, as in GRUs *update* gate) will favor items in the context/cell state (transformed through a *tanh* function to simulate memory decay).

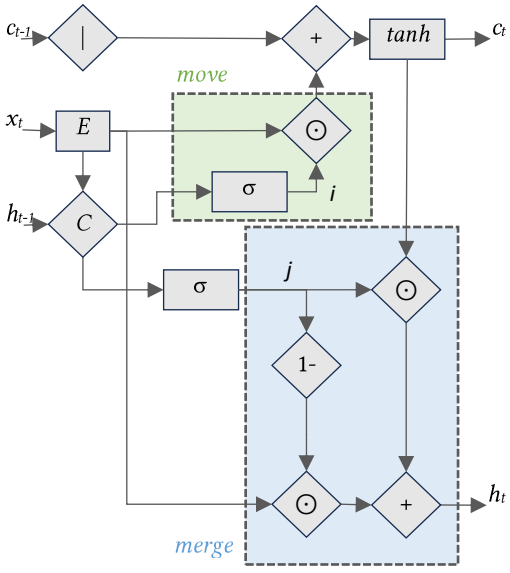

Figure 2: eMG-RNN cell computational graph.

This architecture is the most performant compared to various alternatives tested for the BabyLM 2024 challenge [32].

2.4. A linguistically informed evaluation

The last step in the pipeline requires a linguistically advanced set of oppositions to verify that the structural generalizations can be captured coherently. We adopted the lm-eval package [33] and we included a specific task based on English BLiMP [12]. Most of the contrasts are derived from the COnVERSA test [4]. They consist of minimal pairs ordered following an increasing complexity metric that considers the number of operations necessary to establish a dependency and the locality of such dependency. The examples below illustrate this point by comparing a local agreement dependency with, (3).b, or without, (3).a, a (linear) intervener and a more complex dependency that requires to process an object relative clause (4):

(3) a. Il piatto è pieno. Vs. Il piatto è piena.
the dish.S.M is full.S.M … full.S.F
b. Il muro della casa è rosso
the wall.S.M of the house is red.S.M
Vs. Il muro della casa è rossa.
the wall.S.M of the house is red.S.F

(4) Ci sono due maestri. Uno insegna ed è ascoltato dagli studenti, l'altro si riposa. Quale maestro insegna? *There are two teachers. One teaches and he's listened to by the students, the other rests. Which one teaches?*
Quello che gli studenti ascoltano.
The one who the students listen to
Vs. Quello che ascolta gli studenti.
The one who listens to the students

Four kinds of dependency (agreement, thematic role assignment, pronominal forms usage, questions formation and answering) are considered for a set of 32 distinct syntactic configurations (a total of 344 minimal pairs to be judged, [4]).

3. Materials and Methods

We trained our models on the IUSS High-Performance Cluster with 2 GPU nodes, each with 4 A100 NVIDIA devices and 1T RAM. Each network has been trained with the full corpus using various batched strategies. (i) *Naturalistic*, line-by-line, single exposure to each sentence in the corpus (each epoch corresponds to an exposure of about 3M tokens); (ii) *Conversational*, two sequential lines are used for the input, that is, [line 1, line 2], [line 2, line 3], etc. are batched; this guarantees that a minimal conversational context for each sentence is provided. In this case, each epoch corresponds to an exposure of 6M tokens; (iii) *fixed sequence length*, considering the average sentence length of 54 words per sentence, a window of 60 tokens is used, that is, [tok_1, tok_2 … tok_60], [tok_2, tok_3 … tok_61] … are batched; with this regimen, each epoch corresponds to an exposure of 180M tokens. Roughly speaking, the bare amount of data processed by a 7 y.o. child ranges from 7 to 70M tokens, [34], then training the networks with a *naturalistic* or *conversational* regimen for 3-10 epochs would result in a comparable exposure. We trained the

networks using torch.optim.lr_scheduler (*step_size*=5, *gamma*=0.1) and Adam optimizer (*lr*=0.001) with 16-bit automatic mixed-precision to speed up the (parallel) training for a maximum of 100 epochs. The networks have been implemented in PyTorch (v2.3.1), wrapped in Transformers structures (4.42.4) to maximize compatibility in the lm-eval (v.0.4.3) environment. CUDA drivers v.12.4 were used. The most relevant configurations tested are discussed in the next session.

3.1. Configurations tested

Three different tokenization strategies (BPE, FastWordPiece, and MorPiece) are compared using the best-performing LSTM network [35] , which consists of 650 units for the embedding layer and 650 nodes for each of the two hidden layers. Five different network architectures are compared, with the GroNLP GPT-2-small pretrained model [36] constituting our "top LLM performer". This model was re-adapted to Italian from the GPT-2 English trained model, which was originally trained on approximately 10 billion token corpus, namely various orders of magnitude bigger than our corpus. We then trained on our corpus a comparable bidirectional transformer (BERT), two LSTM networks, respectively with 1 and 2 LSTM layers, and a one-layer eMG-RNN network (Table 2), as described in §2.3.

Table 2
Network architectures

Model	Parameters	Structure
GroNLP GPT-2 small	121M	12 Attention heads + 768 hidden units
BERT	113M	12 Attention heads + 768 hidden units
LSTMx2	65M	650 Embedding + 2 LSTM layers (650)
LSTMx1	36M	650 Embedding + 1 LSTM layers (650)
eMG-RNN	73M	650 Embedding + 1 eMG-RNN layer (650)

4. Results

Comparing BERT and LSTM architectures, LSTMx1 qualifies as the most performant configuration (both in training and in minimal pair judgments). Considering training, the only batching regimen performing sufficiently well is the *fixed sequence length* (*loss=0.8877* with LSTMx1 vs. conversational *loss*=4.0240 or naturalistic regimen *loss*=4.5884). All networks reached a learning plateau around 10-12 epochs. Comparing the performances on COnVERSA, we realized that the results does not improve after 3 epochs of *fixed sequence length* (60 tokens) training regimen (this result is compatible with the overfitting hypothesis, [37]). Focusing on tokenizer training results with LSTMx1, we observed that BPE and FastWordPiece have comparable performance. MorPiece performs slightly worse, even though the tokenization seems linguistically more coherent (e.g., "farlo" – "to do it" is tokenized both by BPE and fWP as a single token, while it is split in two in MorPiece: "far" "+lo") and the training faster (Table 3). This, however, only marginally impacts on minimal pairs contrast judgments, performing slightly better, overall, just in certain agreement cases.

Table 3
Impact of the tokenization strategy on LSTM training

Strategy	Vocab size	Training time x epoch	Loss
Corpus types	*72931*	*~1h*	*1.1520*
BPE	96028	*~4h*	*0.8877*
fWP	97162	*~4h*	*0.9491*
MoP	55049	*~3h*	*1.1151*

We then adopted the BPE tokenizer for architectural comparisons. Network training performances are summarized in Table 4 and graphically represented in Figure 3 for linguistic dimensions comparison.

Table 4
Network architectures and their performance on training (Loss/Accuracy) and COnVERSA test

Model	Loss/Accuracy	COnVERA
GroNLP GPT-2s		0.73 (±0.02)
BERT	4.5488/0.65471	0.43(±0.02)
LSTMx2	0.7849/0.8283	0.48(±0.03)
LSTMx1	0.8784/0.8103	0.52(±0.03)
eMG-RNN	0.9491/0.7815	0.61(±0.01)

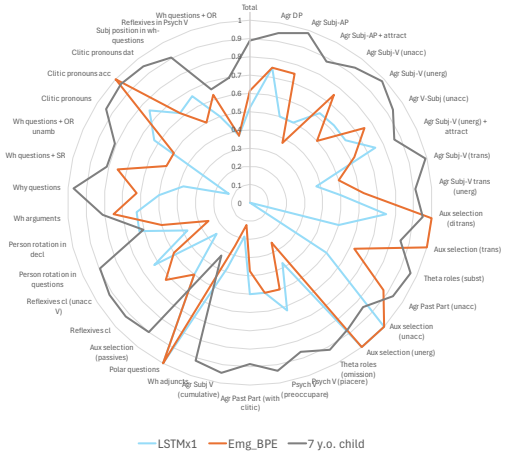

Figure 3: Performance of the 2 best RNN networks variants on COnVERSA compared to the 7 y.o. children.

5. Discussion

Overall, LSTM networks significantly outperform Bidirectional Transformers in this minimal pairs test on Italian. This finding is consistent with results previously discussed in the literature and suggests a clear advantage of recurrent, sequential model architectures (e.g., LSTM) over Bidirectional Transformers in terms of linguistic generalizations [38] and partially justify the renewed interest for RNN networks that we have been observed in the last couple of years [24], [26]. As far as the tokenization procedure is concerned, it is somewhat premature to draw definitive conclusions from our experiments, as MorPiece has not yet been fully optimized or tested. Specifically, the optimal cut-off threshold and minimum branching factor have not been systematically evaluated. Nevertheless, a more morphologically coherent segmentation is expected to enhance sensitivity in certain minimal contrasts.

Similarly, the eMG-RNN architecture could be further explored and optimized, particularly considering specific contrasts, which may help determine whether our linguistic modeling is on the right track. Evidence to the contrary is attested by the judgments of sentences with missing thematic roles, which are often incorrectly preferred by most models, including our eMG-RNN.

In the end, our results suggest that Loss/Accuracy performance registered in training is not a significant predictor of the performance on the COnVERSA test, or more generally, of the linguistic coherence of the LM trained. Likewise, the models' dimension is not a clear predictor either: Transformers trained on the same small dataset perform randomly (in all dimensions their performance is round 50%) while eMG-RNN, which has a number of parameters similar to LSTM-2, outperforms both LSTM-2 and LSTM-1 (half size of eMG-RNN). The training size remains a striking difference compared to the input received by children: this difference of one order of magnitude suggests that the bias considered in eMG-RNN are not yet satisfactory and that our Language Acquisition Device is still more efficient; in this sense, the Poverty of Stimulus Hypothesis remains unrefuted [39] by these results. Next steps will consider extending to 10M tokens the training corpus (to match the English counterpart [1]) and further exploring the effects of optimized tokenization procedures or other minimal modifications, and optimizations [24], of recurrent neural networks.

Acknowledgments

This project is partially supported by the T-GRA2L: Testing GRAdeness and GRAmmaticality in Linguistics, PRIN 2022 Next Generation EU funded Project (202223PL4N). National coordinator: CC

References

[1] A. Warstadt *et al.*, Eds., *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*. Singapore: Association for Computational Linguistics, 2023. [Online]. Available: https://aclanthology.org/2023.conll-babylm.0

[2] R. Katzir, "Why large language models are poor theories of human linguistic cognition. A reply to Piantadosi (2023)," 2023. [Online]. Available: lingbuzz/007190

[3] S. Piantadosi, "Modern language models refute Chomsky's approach to language," *Lingbuzz Preprint, lingbuzz*, vol. 7180, 2023.

[4] C. Chesi, G. Ghersi, V. Musella, and D. Musola, *COnVERSA: Test di Comprensione delle Opposizioni morfo-sintattiche VERbali attraverso la ScritturA*. Firenze: Hogrefe, 2024.

[5] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017, Accessed: Mar. 26, 2022. [Online]. Available: http://arxiv.org/abs/1706.03762

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] L. G. G. Charpentier and D. Samuel, "Not all layers are equally as important: Every Layer Counts BERT," in *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, Singapore: Association for Computational Linguistics, 2023, pp. 210–224. doi: 10.18653/v1/2023.conll-babylm.20.

[8] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," Jun. 23, 2022, *arXiv*: arXiv:2205.14135. Accessed: Jun. 12, 2024. [Online]. Available: http://arxiv.org/abs/2205.14135

[9] J. Steuer, M. Mosbach, and D. Klakow, "Large GPT-like Models are Bad Babies: A Closer Look at the Relationship between Linguistic Competence and Psycholinguistic Measures," in *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, Singapore: Association for Computational Linguistics, 2023, pp. 114–129. doi: 10.18653/v1/2023.conll-babylm.12.

[10] Y. Zhang, A. Warstadt, H.-S. Li, and S. R. Bowman, "When Do You Need Billions of Words of Pretraining Data?," Nov. 10, 2020, *arXiv*: arXiv:2011.04946. Accessed: Jan. 10, 2024. [Online]. Available: http://arxiv.org/abs/2011.04946

[11] S. Crain and M. Nakayama, "Structure Dependence in Grammar Formation," *Language*, vol. 63, no. 3, p. 522, Sep. 1987, doi: 10.2307/415004.

[12] A. Warstadt *et al.*, "BLiMP: The Benchmark of Linguistic Minimal Pairs for English," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 377–392, Dec. 2020, doi: 10.1162/tacl_a_00321.

[13] I. Alfano, F. Cutugno, A. De Rosa, C. Iacobini, R. Savy, and M. Voghera, "VOLIP: a corpus of spoken Italian and a virtuous example of reuse of linguistic resources," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds., Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 3897–3901. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/906_Paper.pdf

[14] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *arXiv:2005.14165 [cs]*, Jul. 2020, Accessed: Apr. 21, 2021. [Online]. Available: http://arxiv.org/abs/2005.14165

[15] P. Gage, "A new algorithm for data compression," *C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] C. D. Yang, *The price of linguistic productivity: how children learn to break the rules of language*. Cambridge, MA: MIT Press, 2016.

[18] H. Jabbar, "MorphPiece : A Linguistic Tokenizer for Large Language Models," Feb. 03, 2024, *arXiv*: arXiv:2307.07262. Accessed: Jun. 23, 2024. [Online]. Available: http://arxiv.org/abs/2307.07262

[19] E. Fredkin, "Trie memory," *Commun. ACM*, vol. 3, no. 9, pp. 490–499, Sep. 1960, doi: 10.1145/367390.367400.

[20] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast WordPiece Tokenization," Oct. 05, 2021, *arXiv*: arXiv:2012.15524. Accessed: Jun. 13, 2024. [Online]. Available: http://arxiv.org/abs/2012.15524

[21] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Sep. 02, 2014, *arXiv*: arXiv:1406.1078. Accessed: Jun. 12, 2024. [Online]. Available: http://arxiv.org/abs/1406.1078

[22] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy: IEEE, 2000, pp. 189–194 vol.3. doi: 10.1109/IJCNN.2000.861302.

[23] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.

[24] L. Feng, F. Tung, M. O. Ahmed, Y. Bengio, and H. Hajimirsadegh, "Were RNNs All We Needed?," Oct. 04, 2024, *arXiv*: arXiv:2410.01201. Accessed: Oct. 18, 2024. [Online]. Available: http://arxiv.org/abs/2410.01201

[25] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," Feb. 22, 2019, *arXiv*: arXiv:1804.07461. Accessed: Jul. 20, 2024. [Online]. Available: http://arxiv.org/abs/1804.07461

[26] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," May 31, 2024, *arXiv*: arXiv:2312.00752. Accessed: Oct. 20, 2024. [Online]. Available: http://arxiv.org/abs/2312.00752

[27] T. Reinhart, "The syntactic domain of anaphora," Massachusetts Institute of Technology, Cambridge (MA), 1976.

[28] J. R. Ross, "Constraints on variables in syntax.," MIT, Cambridge (MA), 1967.

[29] C. Cecchetto, "A Comparative Analysis of Left and Right Dislocation in Romance," *Studia Linguistica*, vol. 53, no. 1, pp. 40–67, Apr. 1999, doi: 10.1111/1467-9582.00039.

[30] N. Chomsky *et al.*, *Merge and the Strong Minimalist Thesis*, 1st ed. Cambridge University Press, 2023. doi: 10.1017/9781009343244.

[31] C. Chesi, "Expectation-based Minimalist Grammars," *arXiv:2109.13871 [cs]*, Sep. 2021, Accessed: Nov. 02, 2021. [Online]. Available: http://arxiv.org/abs/2109.13871

[32] C. Chesi *et al.*, "Different Ways to Forget: Linguistic Gates in Recurrent Neural Networks," in *Proceedings of the BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, 2024.

[33] L. Gao *et al.*, "A framework for few-shot language model evaluation." Zenodo, Dec. 2023. doi: 10.5281/zenodo.10256836.

[34] B. Hart and T. R. Risley, "American parenting of language-learning children: Persisting differences in family-child interactions observed in natural home environments.," *Developmental Psychology*, vol. 28, no. 6, pp. 1096–1105, Nov. 1992, doi: 10.1037/0012-1649.28.6.1096.

[35] K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, and M. Baroni, "Colorless Green Recurrent Networks Dream Hierarchically," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1195–1205. doi: 10.18653/v1/N18-1108.

[36] W. de Vries and M. Nissim, "As Good as New. How to Successfully Recycle English GPT-2 to Make Models for Other Languages," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 836–846. doi: 10.18653/v1/2021.findings-acl.74.

[37] F. Xue, Y. Fu, W. Zhou, Z. Zheng, and Y. You, "To Repeat or Not To Repeat: Insights from Scaling LLM under Token-Crisis," 2023, *arXiv*. doi: 10.48550/ARXIV.2305.13230.

[38] E. Wilcox, R. Futrell, and R. Levy, "Using Computational Models to Test Syntactic

Learnability," *Linguistic Inquiry*, pp. 1–44, Apr. 2023, doi: 10.1162/ling_a_00491.

[39] C. Yang, S. Crain, R. C. Berwick, N. Chomsky, and J. J. Bolhuis, "The growth of language: Universal Grammar, experience, and principles of computation," *Neuroscience & Biobehavioral Reviews*, vol. 81, pp. 103–119, Oct. 2017, doi: 10.1016/j.neubiorev.2016.12.023.

A. Online Resources

Resources (corpus information, tokenizer, network architectures and lm_eval tasks) are available at https://github.com/cristianochesi/babylm-2024.