A surprisal oracle for active curriculum language modeling

Xudong Hong[‡], **Sharid Loáiciga**[†] **and Asad Sayeed**[†]

[‡]Dept. of Language Science and Technology and Dept. of Computer Science, Saarland University
[†]Dept. of Philosophy, Linguistics, and Theory of Science, University of Gothenburg
{xhong}@lst.uni-saarland.de, {sharid.loaiciga, asad.sayeed}@gu.se

Abstract

We investigate the viability of surprisal in an active curriculum learning framework to train transformer-based language models in the context of the BabyLM Challenge. In our approach, the model itself selects the data to label (active learning) and schedules data samples based on a surprisal oracle (curriculum learning). We show that the models learn across all the tasks and datasets evaluated, making the technique a promising alternative approach to reducing the data requirements of language models. Our code is available at https://github.com/asayeed/ActiveBaby.

1 Introduction

We describe our submission to the BabyLM Challenge (Warstadt et al., 2023), a shared-task about language models trained from scratch on a developmentally plausible corpus. Inspired by expectation-based theories of sentence processing (Hale, 2001; Levy, 2008) and active curriculum learning (ACL) (Jafarpour et al., 2021), our approach relies on surprisal to select informative samples and streamline them into the model during training. We henceforth refer to our strategy as active curriculum learning modeling (ACLM).

There is a large volume of published studies describing how the processing difficulty of a sentence is correlated with its incremental probability in context (Linzen and Jaeger, 2016; Futrell and Levy, 2017; Hahn et al., 2019, among others). In other words, as people process sentences, they generate predictions about what is coming next and this can be measured using surprisal (Demberg et al., 2012). Here, we test to what extent this principle of syntactic predictability can also be used to guide the learning of a language model.

ACL, on the other hand, combines the strengths from Active Learning (AL) and Curriculum Learning (CL). AL is a classic paradigm for small data supervised scenarios, whereby an oracle labels informative examples selected by the model itself based (most often) on a uncertainty heuristic. The uncertainty metrics, however, tend to bias the model towards eccentric examples (Zhang et al., 2022b). To counteract this, Jafarpour et al. (2021) use CL, a technique that mimics how humans learn by regulating the training according to some schedule criterion, e.g., easy to difficult or short to long examples (Bengio et al., 2009).

In our approach, we use surprisal as sampling heuristic. A sample is formed from the sentence with the highest surprisal value s from an initial pool, along with the n most similar sentences to s from the rest of the training data. At each iteration, a new sample is added to the pool until convergence.

Our results show that the technique successfully learns steadily and incrementally in all the tasks, although its performance remains modest in comparison with equivalent systems with full access to the training data.

2 Background

AL specifically aims at reducing the amount of examples required for training. In AL, it is the algorithm itself that selects the most informative examples to annotate based on a probabilistic query heuristic. Each example is used to make the model better at selecting the next example. Nevertheless, AL is difficult to implement with neural networks frameworks due to their large number of parameters leading to poor uncertainty estimation and model instability (Lowell et al., 2019; Schröder et al., 2022). An excellent survey about the latest work on AL specifically for NLP is presented by Zhang et al. (2022b).

There is remarkably little research on surprisal and AL, or surprisal and CL. In the context of sentence classification, Yuan et al. (2020) exploit a pre-trained BERT model (Devlin et al., 2019) to generate surprisal embeddings as input to the

Proceedings of the 27th Conference on Computational Natural Language Learning: Volume 2: The BabyLM Challenge, pages 259–268
December 6-7, 2023

sentence labeling part of their model. In our case, sentence surprisal is used to select the sentence seeding the samples and the model is trained with a language modeling objective. Similar ideas are found in the context of machine translation.

Zhang et al. (2021) have experimented with adding training samples from a pool based on a difficulty criterion operationalized as sentence length (short sentences are easy, long ones are difficult) and word rarity (common sentences are easy, rare ones are difficult). In the second case, rare words are estimated based on the logarithms of word probabilities averaged over the sentence, which is effectively the same as surprisal. Likewise, Zhou et al. (2021) also report sampling based on sentence length and word rarity. In addition, they experiment with the probability of the sentence from an independent language model, source sentence word embeddings from another independent model, and the sentence score of the model under training itself. Last, Mohiuddin et al. (2022) rank their training sentences from easy to hard using the prediction scores of the model under training. They experiment with different window ranges over the distribution of these scores.

In keeping with the goals of the shared task, we train a language model from scratch. Elsewhere, a considerable amount of literature has been published on *compressing* state-of-the-art large language models (LLMs) into much smaller models without losing too much in accuracy and performance (Sanh et al., 2020; Zhang et al., 2022a, among others).

Cognitive studies, on their part, use LLMs to predict estimates about different effects attested in human language processing (Linzen et al., 2016; Futrell and Levy, 2019; Wei et al., 2021). This type of work also sheds light on the biases and mechanisms of learning of the LLMs themselves. Sinha et al. (2021), for instance, find the LLMs can account for word order due to their capacity for higher-order word co-occurrence statistics, while Arehalli et al. (2022) and Oh and Schuler (2023) have raised questions about the reliability of LLMs predictions due to their conflation of lexical and syntactic biases and their large capacity to memorize linguistic structures.

Humans acquire language in the context of interaction with a social and physical environment, which may explain at least part of the inductive bias humans display that allows them to learn from quantities of data far less than LLMs typically require to produce some of the spectacular-seeming recent results. The `strict` and `strict-small` settings of the BabyLM challenge effectively probe how small we can make the training data in an ungrounded setting. In this context, we still hypothesize that an interactive, environment-aware approach will be important in making learning efficient. We conceive of the learner as seeking out stimuli that represent domains of syntax and semantics on which the learner is furthest away from convergence, and we represent that distance by surprisal. We then hypothesize that the learner is motivated to seek out or pay attention to items that have a similar pattern of overall uncertainty, even if the specific syntactic or semantic conditions may be different in terms of, e.g., parts of speech or lexical semantics.

3 The model

Training a model with active learning (Cohn et al., 1996) involves (1) selecting an initial training set of sentences from a pool of sentences available for future training iterations and (2) iteratively adding sentences from the pool to the training set based on a criterion of uncertainty about the data. For classification tasks in scenarios with limited labelled data, this involves a human in the loop who labels a selection of "least certain" data from the pool, where the certainty is calculated based on model confidence. This form of active learning is intended to reduce the difficulty of labelling training data when, for example, annotators are difficult to find—only label what the model finds most "interesting" for the learning algorithm. This concept can be extended from classification to, for example, machine translation in low-resource contexts (Gupta et al., 2021), where a small group of proficient translators would be prompted for translations of items in the pool that the model is, e.g., most perplexed about.

Pre-training a language model is, however, not primarily a classification task. For a generative language model, the learning goal is for the model to be able to produce the next token or set of tokens given a prefix and to do so until a complete utterance is produced. Uncertainty for a generative LM over an utterance requires the aggregate of uncertainty over a number of decisions, each with low prior probability. Insofar as the model is intended to represent an approximation of human acquisition, it is implausible that the pool (representing

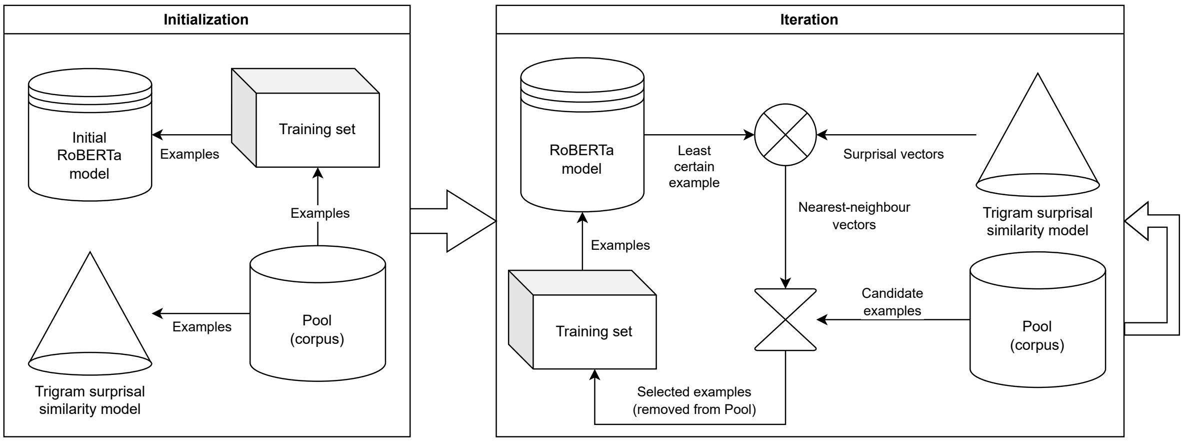

Figure 1: The architecture of our ACLM method.

the full environment over time of the learner) be fully evaluated in advance for uncertainty in the service of training data selection. This requires the introduction of an additional criterion for selecting new examples that are likely to represent utterances that are currently uncertain to the model.

To solve this, we adapt the concept of Active Curriculum Learning (ACL) from Jafarpour et al. (2021), who envision a joint scoring criterion for the selection of additional examples, composed of the scoring criterion for an active learning algorithm and the scoring criterion for a curriculum learning algorithm. Our approach is two-step, rather than a linear combination of two criteria. In the first step, we use a trained model to select the least certain example from the *existing* training set, rather than the pool. Then we apply a heuristic to select sentences that are structurally similar to the current least certain training example and add them to the next iteration's training set (see Figure 1).

Our heuristic is similarity based on a profile of the token-by-token incremental trigram surprisal of each sentence. Profiles of all the training and pool sentences are represented as seven-dimensional surprisal vectors by rescaling the sequence of surprisal values, which varies by the sentence length. This enables us to take the least certain training example's surprisal vector and request the nearest-neighbours, which are then added to the training set.

3.1 Base model

The base model is RoBERTa (Liu et al., 2019; Zhuang et al., 2021) trained from initialization on a 100K randomly selected subset—the initial training set—of the `strict-small` dataset of the BabyLM challenge.

The data for all our model variants was preprocessed in the same way. The documents where split at the sentence level and then BPE tokenized with a truncated maximum length of 512 tokens.

3.2 Surprisal space

The surprisal space for the corpus as a whole is generated by training a simple language model via Maximum Likelihood Estimation on n-grams up to trigrams via the nltk.lm module. Trigram surprisal can be used to explain part of human linguistic behaviour at a syntactic and semantic level in human dialogue (Sayeed et al., 2015).

Every sentence in the pool and training set is then labelled with a sequence of surprial values, one for each token. We use scikit-image's resizing function to stretch or shrink the surprisal sequences to vectors of dimension seven.[1]

All the vectors are placed in an instance of scikit-learn's KDTree (Sproull, 1991) implementation, which allows for an efficient search for the k nearest neighbours (kNN) of a given query vector and returns sentence identifiers for the vectors in the pool that are nearest to the surprisal vector of the least certain example. These are added to the training set.

For efficiency reasons, we do not re-evaluate the surprisal space at every iteration of active learning. This part of the model represents an oracle selecting items from the pool that bear a model uncertainty pattern that is similar to the least certain item in the training set.

[1]This is a random choice to get a small number such that the surprisal space can fit into the main memory.

3.3 Active curriculum language modeling

RoBERTa is allowed to train with the current training set for multiple epochs until the least certain training set example is found and the active learning loop initiated. This process thus combines active learning, in terms of the model being used to identify sets of data that need to be labelled, and curriculum learning, where a heuristic—a vector-based surprisal oracle—is used to schedule the newly delivered examples. We stop the model training after a set number of iterations.

The least certain example is the one with the highest cross-entropy loss or surprisal according to the model; that is, while the surprisal vectors do not change between iterations based on the RoBERTa model, the model under training changes to produce a different ranking of sentences in its training set, thereby allowing for variation in curriculum presented by the surprisal oracle.

4 Results

4.1 Shared task evaluation

We use the official evaluation tools (Gao et al., 2021) from the BabyLM Challenge to report our results. Our submissions mostly targeted the `strict-small` track, but we also report results for one system trained for the `strict` track. Tables 1, 2 and 3 in Appendix A contain the details of the obtained scores.

Strict-100M is trained with the data from the `strict` track, all other models rely on the `strict-small` data. 10ep10it and 10ep20it served as our internal baselines. They are RoBERTa models without ACLM that only differ in the number of iterations, 10 for the first and 20 for the second, both have a batch size of 64 sentences. The ACLM models are s50Kep1 and s50Kep5. Both have a batch size of 64 and use a sample size of 50K sentences; they differ in that the first runs one epoch per sample and the second 5 epochs per sample.

In summary, the results for the Strict-100M model tend to be overall higher, as it is trained on a larger amount of data. When considering the ACLM models, we observe that they performed the best when evaluated on the (Super)GLUE datasets and the worst on the MSGS one. There is also a clear gain in performance when training the model with more epochs per sample.

Figure 2: Comparison of the learning curves of systems with random sampling (green line), sampling with maximal surprisal (orange line), and sampling with minimal surprisal criterion (red line).

4.2 Hyper-parameter search

We experimented with batch sizes of 32 and 64 data points and observed that it produced minimum differences. As for the number of epochs, we tested different values between 1 and 5 for the ACLM systems, with 5 yielding the best performance. We expected to see some variation if changing the size of the sample size, but we also did not observe any important changes.

5 Analysis

5.1 Sampling Methods

Our method set out to determine the extent to which the principle of predictability as represented by surprisal can be used to guide language model training. In order to test this hypothesis, we compared the best performing ACLM system (s50Kep5) using three different values of surprisal for the query: minimum, maximum, and random (Figure 2). What we found is that the model with the maximal surprisal performed closely to the random one and learned faster, while the one with minimal surprisal did clearly well on evaluation. While this

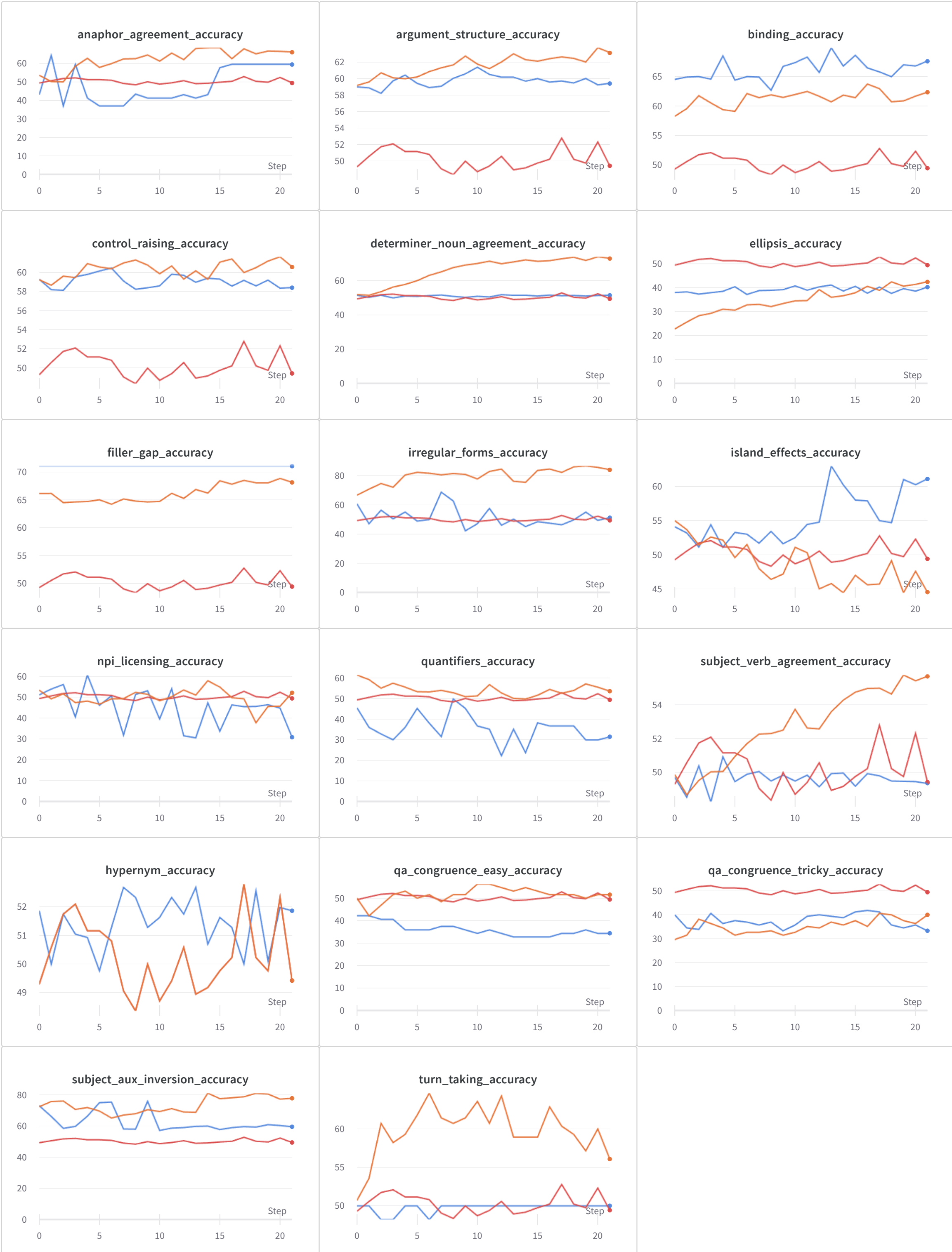

Figure 3: Accuracy of the systems 10ep10it (blue line, without ACLM), 50Kep5min (red line, with ACLM and minimal surprisal sampling) and s50K_ep5 (orange line, with ACLM) in the zero-shot tasks over 20 checkpoints during training.

seemed counter-intuitive at first, we believe that the model with the minimal surprisal is actually selecting sentences that are overall more informative than those with the maximal surprisal which might be too divergent. Furthermore, this also accords with Mohiuddin et al.'s (2022) analysis that if a sample is too easy, the model might not gain any useful information from it, whereas if the sample is too hard, it might degrade the model's performance at that point. Taken together, this strongly suggests that surprisal does have an effect as a sampling query, but more work will need to be done to determine the optimal curriculum for its efficiency.

5.2 Zero-shot tasks

As a means to understand the way in which the ACLM models learn, we evaluated the 20 training checkpoints of the models 10ep10it, 50Kep5 and 50Kep5min (50Kep5 which samples data points with minimal surprisal) on the official zero-shot tasks. As mentioned, while all systems are trained on the `strict-small` data, the 10ep10it system uses all the data at once, in the standard way, while 50Kep5 and 50Kep5min are trained through ACLM with different sampling methods. These systems have a sample size of 50k sentences and runs 5 epochs per sample. Both have a batch size of 64. Results are depicted in Figure 3.

The plots from this figure indicate that the ACLM model learns in a steadier fashion than its non-ACLM counterpart, in particular for the "agreement" categories: determiner-noun, subject-verb and (somewhat less) anaphor agreement. This might indicate a frequency effect better caught on by the ACLM model, as basically every sentence contains a positive example of correct agreement, but it is unknown how many total examples there are of the other tested phenomena. For most of the other categories, the learning curves are similar overall, and the ACLM model shows consistent learning increments. The exception seems to be the island effects category, where the accuracy tends to drop over time. Surprisingly, the ACLM model with minimal surprisal sampling (50Kep5min) underperforms the ACLM model with maximal surprisal (50Kep5) across many tasks except congruence-tricky and island, effects even though 50Kep5min has a lower evaluation loss than 50Kep5. The results indicate that maximal surprisal sampling is an effective method to improve model performance on zero-shot grammatical tasks. Moreover, lower perplexity does not always imply better performance on linguistic tasks.

6 Conclusions and future work

To our knowledge, this is the first contribution to the literature in reducing the pre-training requirement of a transformer-based language model via active curriculum learning modeling. What we have shown is that learning does take place under these conditions and produces promising results. It is not the case, however, that we explored the full potential of this technique; there is a huge scope for plausible variants that may be even more effective than what we have proposed.

For example, we designed the surprisal oracle around a vector space defined by trigram surprisal over tokens which is never re-evaluated. A more realistic learner would re-evaluate the surprisal space based on what it knows now, i.e., compute per-token surprisal based on the current training state of the transformer model. We did not implement this for computational resource reasons.

Another likely possibility for improvement of our model lies in the fact that the surprisal space is created by resizing all the vectors to the same dimensionality, which is equivalent to representing all sentences as having the same length. It is implausible that longer sentences produce model uncertainty in the same way as shorter sentences. A future version of our work could attempt to bin the sentences by length, creating separate surprisal spaces.

Limitations

The models trained in this study are designed to test ACLM as a viable method to train language models and as such, they are not overly optimized. Furthermore, any claims are specific to English, in keeping with the shared-task constraints.

Acknowledgements

This research was funded in part by the Swedish Research Council (VR) grant (2014-39) for the Centre for Linguistic Theory and Studies in Probability (CLASP). Xudong Hong was funded by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA). We thank our student assistant Mattes Alexander Warning for searching hyperparameters for our models. We also thank the anonymous reviewers for their insightful comments.

References

Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. In *Proceedings of the 26th*

Conference on Computational Natural Language Learning (CoNLL), pages 301–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

Vera Demberg, Asad Sayeed, Philip Gorinski, and Nikolaos Engonopoulos. 2012. Syntactic surprisal affects spoken word duration in conversational contexts. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 356–367, Jeju Island, Korea. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Futrell and Roger Levy. 2017. Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 688–698, Valencia, Spain. Association for Computational Linguistics.

Richard Futrell and Roger P. Levy. 2019. Do RNNs learn human-like abstract word order preferences? In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 50–59.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Kamal Gupta, Dhanvanth Boppana, Rejwanul Haque, Asif Ekbal, and Pushpak Bhattacharyya. 2021. Investigating active learning in interactive neural machine translation. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 10–22, Virtual. Association for Machine Translation in the Americas.

Michael Hahn, Frank Keller, Yonatan Bisk, and Yonatan Belinkov. 2019. Character-based surprisal as a model of reading difficulty in the presence of errors. In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, pages 401–407. cognitivesciencesociety.org.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active curriculum learning. In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 40–45, Online. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Tal Linzen and Florian T. Jaeger. 2016. Uncertainty and Expectation in Sentence Processing: Evidence From Subcategorization Distributions. *Cognitive science*, 40(6):1382–1411.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30, Hong Kong, China. Association for Computational Linguistics.

Tasnim Mohiuddin, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty. 2022. Data selection curriculum for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1569–1582, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023. Why Does Surprisal From Larger Transformer-Based Language Models Provide a Poorer Fit to Human Reading Times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Asad Sayeed, Stefan Fischer, and Vera Demberg. 2015. Vector-space calculation of semantic surprisal for predicting word pronunciation duration. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International*

Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 763–773, Beijing, China. Association for Computational Linguistics.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert F Sproull. 1991. Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica*, 6:579–589.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Adina Williams, Bhargavi Paranjabe, Tal Linzen, and Ryan Cotterell. 2023. Findings of the 2023 BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 BabyLM Challenge*. Association for Computational Linguistics (ACL).

Jason Wei, Clara Meister, and Ryan Cotterell. 2021. A cognitive regularizer for language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5191–5202, Online. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Mingliang Zhang, Fandong Meng, Yunhai Tong, and Jie Zhou. 2021. Competence-based curriculum learning for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2481–2493, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minjia Zhang, Niranjan Uma Naresh, and Yuxiong He. 2022a. Scala: Accelerating adaptation of pre-trained transformer-based language models via efficient large-batch adversarial noise.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022b. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lei Zhou, Liang Ding, Kevin Duh, Shinji Watanabe, Ryohei Sasano, and Koichi Takeda. 2021. Self-guided curriculum learning for neural machine translation. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 206–214, Bangkok, Thailand (online). Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

	Submitted RoBERTa models					Official baselines		
		Strict small 10M						
				ACL				
	Strict-100M	10ep10it	10ep20it	s50Kep1	s50Kep5	OPT-125m	RoBERTa-base	T5-base
Anaphor Agr.	82.31	77.76	74.34	42.02	75.30	63.8	**81.5**	68.9
Agr. Structure	74.03	**72.91**	68.83	61.52	60.36	70.6	67.1	63.8
Binding	68.63	69.09	67.62	64.02	**85.95**	67.1	67.3	60.4
Control/Raising	70.35	**68.96**	64.98	61.36	50.03	66.5	67.9	60.9
Det-N Agr.	94.84	**95.66**	91.94	55.49	55.79	78.5	90.8	72.2
Ellipsis	65.42	65.82	56.41	32.79	55.41	62	**76.4**	34.4
Filler-Gap	78.32	**75.61**	69.89	63.68	50.12	63.80	63.50	48.20
Irregular Forms	92.01	**89.41**	89.87	75.01	43.98	67.5	87.4	77.6
Island Effects	48.62	46.30	40.58	47.20	**50.00**	48.6	39.9	45.6
NPI Licensing	61.52	54.16	**56.77**	51.90	35.15	46.7	55.9	47.8
Quantifiers	66.82	66.87	63.96	45.96	**78.02**	59.6	70.5	61.2
S-V Agr.	80.85	**79.33**	70.66	50.44	60.39	56.9	65.4	65
				Supplement				
Hypernym	49.07	49.30	49.07	50.23	**62.15**	50	49.4	48
QA Cong. (easy)	57.81	56.25	53.13	50.00	**66.51**	54.7	31.3	40.6
QA Cong. (tricky)	33.33	35.76	35.76	30.30	**69.17**	31.5	32.1	21.2
Subj.-Aux. Inv.	78.92	75.38	**82.73**	75.82	62.03	80.3	71.7	64.9
Turn Taking	57.50	61.79	**66.79**	56.43	42.96	57.1	53.2	45

Table 1: Accuracy scores of the zero-shot evaluation on the BLiMP dataset. Comparisons per row highlighted with bold do not include the Strict-100M column. QA Cong. means QA Congruence. Inv. means inversion.

	Submitted RoBERTa models					Official baselines			
		Strict small 10M							
				ACL					
	Strict-100M	10ep10it	10ep20it	s50Kep1	s50Kep5	Majority	OPT-125m	RoBERTa-base	T5-base
CoLA	73.11	**72.62**	70.76	69.48	61.17	69.5	64.6	70.8	61.2
SST-2	86.42	**84.84**	83.27	81.3	75.97	50.2	81.9	87	78.1
MRPC	63.28	64.41	64.41	64.41	90.2	82	72.5	79.2	80.5
QQP	79.93	81.65	**79.88**	77.65	65.98	53.1	60.4	73.7	66.2
MNLI	69.02	70.34	68.62	65.27	**100**	35.7	57.6	73.2	48
MNLI-mm	71.94	**71.26**	69.51	67.06	66.6	35.7	60	74	50.3
QNLI	64.96	66.4	66.49	58.36	**68.44**	35.4	61.5	77	62
RTE	47.47	51.52	49.49	49.49	**98.93**	53.1	60	61.6	49.4
BoolQ	65.98	63.35	66.11	66.11	**74.9**	50.5	63.3	66.3	66
MultiRC	57.28	58.6	56.19	50.82	58.6	**59.9**	55.2	61.4	47.1
WSC	61.45	61.45	61.45	61.45	**81.89**	53.2	60.2	61.4	61.4

Table 2: Accuracy scores of the fine-tuning evaluation on the (Super)GLUE datasets. Comparisons per row highlighted with bold do not include the Strict-100M column.

	Submitted RoBERTa models					Official baselines		
		Strict small 10M						
				ACL				
	Strict-100M	10ep10it	10ep20it	s50Kep1	s50Kep5	OPT-125m	RoBERTa-base	T5-base
CR (Control)	91.55	86.68	86.89	75.51	**94.5**	86.4	84.1	78.4
LC (Control)	**100**	**100**	**100**	**100**	66.45	86.1	**100**	**100**
MV (Control)	99.72	99.77	99.63	97.57	84.33	**99.8**	99.4	72.7
RP (Control)	98.85	**100**	**100**	97.87	0	**100**	93.5	95.5
SC (Control)	81.27	89.54	90.54	88.17	66.78	94.3	**96.4**	94.4
CR_LC	66.76	66.74	66.69	66.32	**83.46**	66.5	67.7	66.7
CR_RTP	66.78	67.25	66.73	66.61	66.71	67	68.6	**69.7**
MV_LC	66.51	66.61	66.61	66.61	55.1	66.5	**66.7**	66.6
MV_RTP	67.18	69.08	67.04	66.71	**100**	67.6	68.6	66.9
SC_LC	63.83	66.28	67.49	67.44	66.73	80.2	**84.2**	73.6
SC_RP	62.32	65.05	64.86	64.07	66.19	67.5	65.7	**67.8**

Table 3: Accuracy scores of the fine-tuning evaluation on the MSGS datasets. Comparisons per row highlighted with bold do not include the Strict-100M column.