# How Much Do Modifications to Transformer Language Models Affect Their Ability to Learn Linguistic Knowledge?

**Simeng Sun**[1] **Brian Dillon**[2] **Mohit Iyyer**[1]
[1]College of Information and Computer Sciences, University of Massachusetts Amherst
[2]Department of Linguistics, University of Massachusetts Amherst
{simengsun, bwdillon, miyyer}@umass.edu

## Abstract

Recent progress in large pretrained language models (LMs) has led to a growth of analyses examining what kinds of linguistic knowledge are encoded by these models. Due to computational constraints, existing analyses are mostly conducted on publicly-released LM checkpoints, which makes it difficult to study how various factors during *training* affect the models' acquisition of linguistic knowledge. In this paper, we train a suite of small-scale Transformer LMs that differ from each other with respect to architectural decisions (e.g., self-attention configuration) or training objectives (e.g., multi-tasking, focal loss). We evaluate these LMs on BLiMP, a targeted evaluation benchmark of multiple English linguistic phenomena. Our experiments show that while none of these modifications yields significant improvements on aggregate, changes to the loss function result in promising improvements on several subcategories (e.g., detecting adjunct islands, correctly scoping negative polarity items). We hope our work offers useful insights for future research into designing Transformer LMs that more effectively learn linguistic knowledge.

## 1 Introduction

At the core of many natural language processing tasks are language models (LMs), which compute the probability distribution of the next token that follows a given input context. The Transformer (Vaswani et al., 2017), as one of the most popular architectures for language modeling, has been widely adopted for large-scale pre-training, such as in BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020). The success of large-scale LM pretraining has propelled a surge of analysis on the linguistic knowledge encoded by language models.

While prior works have uncovered many exciting facts regarding the linguistic capability of those pretrained LMs (Hewitt and Manning, 2019; Liu et al., 2019; Jawahar et al., 2019), most of these analyses are conducted on publicly-released model checkpoints, and thus the impact of various LM *training* configurations remains relatively unexplored, limited to LSTM LM configurations (Linzen et al., 2016) or varying training data size (Zhang et al., 2021).

In this work, we focus on Transformer LMs (Vaswani et al., 2017) instead of LSTMs, and we investigate two aspects of LM training distinct from previous works – (1) the LM training objective, for which we experiment with the focal loss and multi-task training; and (2) the Transformer's self-attention mechanism, which we restrict to a local window of tokens. We train a suite of Transformer LMs that minimally differ from each other in one of these two aspects, and evaluate the effect of these changes via non-parametric probing on BLiMP (Warstadt et al., 2020a), a targeted evaluation benchmark of multiple English linguistic phenomena (e.g., island effects, anaphor agreement). Experimental results demonstrate that *none* of these modifications yields significant gains on BLiMP *in aggregate*. However, we do observe that modified training objectives (e.g, using focal loss instead of standard cross entropy loss) result in improvements to specific *subtypes* of linguistic phenomena. Overall, our experiments suggest that it could be promising to scale up Transformer LMs with modified training objectives, as they may help improve syntactic generalization.

## 2 Method

Language models compute $p(w_i \mid w_{<i})$, the probability distribution of the next token $w_i$ given the preceding context $w_{<i}$. The conventional training objective of an LM is to minimize the surprisal of tokens in a training set. The surprisal of a single token can be expressed as the negative log probability of that token given the preceding context

*Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 46 - 53
May 26, 2022 ©2022 Association for Computational Linguistics

(prefix):

$$l_i = -\log p(w_i \mid w_{<i})$$

While many models were proposed to compute $p(w_i \mid w_{<i})$, we focus on the Transformer architecture (Vaswani et al., 2017), which consists of a stack of alternated self-attention and feed-forward blocks and has become the mainstream architecture for large-scale LM pretraining.

Unlike prior work, which has focused on *fixed* Transformer language model checkpoints, we are curious to see how intervening in the training process would impact the resulting models. Specifically, we ask: **are there any training objectives or model design choices that would improve the models' acquisition of linguistic knowledge?**

## 2.1 Altered training process

To understand how varying training configurations affect the linguistic capacities of the final models, we narrow our focus to the LM training objective and the self-attention mechanism. We train a set of Transformer LMs, each differing from each other in only the changes described below:

**Focal loss (FL)** As shown by Zhang et al. (2021), language models learn different linguistic phenomena at different speeds and require different amounts of data. For instance, the learning curve for *subject-verb agreement* phenomena plateaus after training on more than 10M tokens, whereas *filler gap dependencies* display steadily increasing performance even up to 30B tokens of training data. This suggests that each phenomenon has an inherent "difficulty", with some requiring more data for an LM to master. In such a scenario, can we improve the acquisition of linguistic knowledge by forcing the model to pay more attention to the "difficult" tokens? To achieve this, one potential alternative to the standard log loss training objective is focal loss (Lin et al., 2018), which can be intuitively explained as reducing the penalty on "easy" well-predicted tokens and increasing the penalty on the "hard" tokens. Formally, the surprisal of each target token is negatively scaled by the predicted probability:

$$l_i^{FL} = -(1 - p(w_i \mid w_{<i}))^{\gamma} \log(p(w_i \mid w_{<i}))$$

Here, $\gamma$ is a hyper-parameter controlling the relative importance between poorly-predicted and well-predicted tokens. Larger values of $\gamma$ allocate more weight to tokens with high surprisal.

**Masked loss (ML)** In the focal loss setting, well-predicted tokens still receive a certain amount of penalty. As an extreme version of the focal loss setting, we simply zero out the loss (masked loss) for the tokens whose predicted probability exceeds a given threshold. Formally, given a threshold $t$, the masked loss is thus:

$$l_i^{ML} = -\Big(1 - \mathbb{I}(p(w_i \mid w_{<i}) \geq t)\Big) \log \Big(p(w_i \mid w_{<i})\Big)$$

**Auxiliary loss (AL)** Multitask training is commonly adopted to provide extra supervision signals to the language model (Winata et al., 2018; Zhou et al., 2019). To explicitly endow an LM with better understanding of syntactic knowledge, we add an auxiliary task where the model is trained to predict labels derived from an external constituency parser using the final layer's token-level representations. The loss of this prediction task is added to the original loss, weighted by a hyper-parameter $\alpha$.

$$l_i^{AL} = -\alpha \log p(w_i \mid w_{<i}) - (1-\alpha) \log p(c_i \mid w_{<i})$$

$c_i$ denotes the linguistic label for each token, which we obtain by associating a token with both the the smallest non-terminal constituent type containing that token and the depth of that constituent in the parse tree. For example, a noun phrase "`red apple`" having depth 3 in the parse tree will have "`NP3 NP3`" as the labels for the auxiliary task.

**Local attention (LA)** Besides the training objective, modifying the architecture is another way to change the inductive biases of the model. As there is a huge number of potential architectural modifications, we constrain our changes to only the attention mechanism as it does not change the total number of parameters and is thus easier to perform a fair comparison. Instead of using the standard self-attention, we adopt *local attention*, where the attention window is limited to only $k$ tokens immediately preceding the target token (Child et al., 2019; Roy et al., 2021; Sun and Iyyer, 2021). We hope that these local attention variants can more easily pick up a recency bias previously shown to exist in RNN language models (Kuncoro et al., 2018). However, note that although the model only attends to the previous $k$ tokens in each layer, the effective receptive field can still be large as the information is propagated through the stacked Transformer layers.

### 2.2 Evaluation on BLiMP

To measure the amount of linguistic knowledge captured by each language model variant, we use BLiMP (Warstadt et al., 2020a), a benchmark of English linguistic minimal pairs. It contains pairs of grammatical and ungrammatical sentences, the latter of which is minimally edited from the grammatical one. The sentence pairs fall into 67 paradigms spanning 12 common English grammar phenomena[1]. A language model makes the correct prediction on this task when it assigns the grammatical sentence higher probability than the ungrammatical one. Each paradigm contains 1K examples, and the accuracy of each paradigm can be treated as a proxy of the amount of specific linguistic knowledge encoded by the LM.

## 3 Experiments

**Data:** We use the same English Wikipedia data used by Gulordava et al. (2018) for our LM pretraining corpus. This corpus contains around 100M tokens in total (80M for training). The vocabulary includes 50K words and a special `<unk>` token substituted for infrequent words.

**Models:** We present four models each trained with slightly different setting. **(1) Focal Loss (FL)**: This model is trained with focal loss, the $\gamma$ is set to 2.[2] **(2) Masked Loss (ML)**: This model is trained with masked loss, with the masking threshold set to $0.9$.[3] **(3) Auxiliary Loss (AL):** This model is trained with auxiliary task of predicting the constituent label, where $\alpha$ is set to 0.5. **(4) Local Attention (LA)**: This is the Transformer in which all self-attentions are replaced with local attention on the preceding 5 tokens.[4]

**Training:** Following prior work on this dataset (Dai et al., 2019; Sun and Iyyer, 2021), we train 16-layer Transformer language models with embedding dimension size 410, hidden dimension 2100, and 10 attention heads per layer. The models are trained with the Adam optimizer $\beta_1 = 0.9, \beta_2 = 0.999$, learning rate 0.00025, and 2000 warmup steps for max 150K steps. Training is performed on GeForce GTX 1080 Ti GPUs and early stopped (average 26h training) when the validation loss stops decreasing for consecutive 10 checkpoints. All evaluations were conducted on model checkpoints with the lowest validation loss.

| **Phenomena** | **BASE** | **FL** | **ML** | **AL** | **LA** |
|---|---|---|---|---|---|
| island | 0.52 | **0.55** | 0.55 | 0.50 | 0.53 |
| anaphor_agree | **0.97** | 0.96 | 0.94 | 0.97 | 0.96 |
| arg_struct | 0.64 | 0.62 | 0.63 | **0.64** | 0.63 |
| det_noun | 0.84 | **0.87** | 0.85 | 0.85 | 0.86 |
| subj_verb | **0.86** | 0.85 | 0.85 | 0.85 | 0.84 |
| ellipsis | 0.76 | 0.79 | 0.77 | 0.76 | **0.81** |
| ctrl_raising | 0.72 | **0.74** | 0.71 | 0.72 | 0.72 |
| quant | 0.70 | 0.69 | 0.68 | 0.64 | **0.71** |
| irregular_form | 0.91 | 0.93 | 0.92 | **0.95** | 0.92 |
| npi | 0.64 | 0.66 | 0.67 | 0.63 | **0.68** |
| binding | 0.75 | 0.76 | 0.75 | **0.77** | 0.76 |
| filler_gap | 0.73 | 0.72 | 0.72 | 0.71 | **0.74** |
| **Average** | 0.75 | 0.76 | 0.75 | 0.75 | **0.76** |

Table 1: Performance of each LM variant on BLIMP, each phenomenon is averaged over subcategories within. **BASE** stands for baseline model, **FL** stands for the model trained with focal loss ($\gamma = 2$), **ML** stands for the model trained with masked loss ($t = 0.9$), **AL** stands for model trained with auxiliary loss, **LA** stands for the model trained with local attention.

## 4 Results & Analysis

Overall, we did not find a significant improvement on BLiMP after applying the aforementioned modifications. Table 1 contains the averaged score of each model evaluated on BLiMP. However, zooming in on each category, we notice significant changes in a subset of paradigms. We observe similar aggregate scores because better performance on certain paradigms are canceled out by worse performance on other paradigms within the same phenomena.[5] In this section, we delineate paradigms showing notable gains compared to the baseline model as shown in Table 2. While we present descriptive observations from the experimental results, more ideal analysis should include mechanistic explanation linking the modifications and the resulting inductive biases, such as those in (Lakretz et al., 2019), which we leave as future work.

[1] We refer the readers to (Warstadt et al., 2020a) for detailed description and the construction process of each paradigm.

[2] $\gamma$ is picked from tuning validation perplexity over $\{0.5, 1, 2\}$

[3] $t$ is picked from tuning over $\{0.85, 0.9, 0.95, 0.999\}$

[4] We tried local $\{2, 3, 5, 10\}$, and 5 yielded the lowest validation perplexity.

[5] Table 3 in Appendix contains results of all 67 paradigms of each model evaluated on BLiMP.

| **Paradigms** | **BASE** | **FL** | **ML** | **AL** | **LA** |
|---|---|---|---|---|---|
| Adjunct island | 0.69 | 0.81 | 0.89 | 0.85 | 0.69 |
| Complex NP island | 0.50 | 0.46 | 0.48 | 0.50 | 0.55 |
| Complext left branch | 0.42 | 0.39 | 0.38 | 0.33 | 0.33 |
| Object extraction | 0.74 | 0.78 | 0.77 | 0.67 | 0.80 |
| Echo question | 0.48 | 0.49 | 0.46 | 0.42 | 0.40 |
| Simple question | 0.34 | 0.41 | 0.37 | 0.31 | 0.41 |
| Subject island | 0.31 | 0.41 | 0.40 | 0.39 | 0.37 |
| Wh. island | 0.66 | 0.63 | 0.62 | 0.55 | 0.71 |
| Det. noun agr. 1 | 0.93 | 0.93 | 0.94 | 0.94 | 0.93 |
| Det. noun agr. 2 | 0.94 | 0.95 | 0.94 | 0.93 | 0.95 |
| Det. noun agr. irregular 1 | 0.84 | 0.83 | 0.84 | 0.84 | 0.83 |
| Det. noun agr. irregular 2 | 0.87 | 0.88 | 0.86 | 0.86 | 0.87 |
| Det. noun agr. w/ adj. 2 | 0.81 | 0.88 | 0.84 | 0.84 | 0.86 |
| Det. noun agr. w/ adj. 1 | 0.84 | 0.88 | 0.85 | 0.84 | 0.86 |
| Det. noun agr. w/ adj. irregular 1 | 0.73 | 0.76 | 0.75 | 0.75 | 0.76 |
| Det. noun agr. w/ adj. irregular 2 | 0.77 | 0.82 | 0.79 | 0.79 | 0.84 |
| Ellipsis 1 | 0.70 | 0.73 | 0.75 | 0.69 | 0.78 |
| Ellipsis 2 | 0.82 | 0.84 | 0.79 | 0.82 | 0.84 |
| Matrix q. npi | 0.14 | 0.17 | 0.26 | 0.15 | 0.22 |
| NPI present 1 | 0.58 | 0.53 | 0.54 | 0.47 | 0.59 |
| NPI present 2 | 0.69 | 0.60 | 0.63 | 0.57 | 0.61 |
| Only NPI licensor present | 0.88 | 0.91 | 0.87 | 0.94 | 0.90 |
| Only NPI scope | 0.66 | 0.79 | 0.82 | 0.77 | 0.84 |
| Sent. neg. NPI | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 |
| Sent. neg. NPI scope | 0.50 | 0.60 | 0.56 | 0.49 | 0.58 |
| Object gap | 0.73 | 0.72 | 0.75 | 0.70 | 0.79 |
| Subject gap | 0.88 | 0.87 | 0.89 | 0.84 | 0.91 |
| Subject gap long dist. | 0.92 | 0.88 | 0.84 | 0.87 | 0.88 |
| No gap vs. that | 0.94 | 0.95 | 0.94 | 0.95 | 0.96 |
| No gap long dist. vs. that | 0.97 | 0.98 | 0.98 | 0.98 | 0.96 |
| Gap vs. that | 0.49 | 0.50 | 0.46 | 0.48 | 0.45 |
| Gap long dist. vs. that | 0.17 | 0.15 | 0.18 | 0.15 | 0.20 |

Table 2: Model performance on subset of BLiMP paradigms, each group of paradigms from top to bottom corresponds to island effect, determiner noun agreement, ellipsis, negative polarity item, and filler gap, respectively[1]. Those values below the baseline accuracy are marked in orange, those above in blue.

**Island Effects** An island is a constituent from which a word cannot be moved, e.g., in *"What was Bill thinking while arguing about news?"*, it is illegal to move *news* out of the island: *"What was Bill thinking news while arguing about?"*. The BLiMP benchmark breaks down island effects to eight paradigms based on the type of islands, and we find all our proposed modifications to the training objective lead to much better accuracy on the targeted pairs of adjunct island and sentential subject island. The model trained with masked loss improves identification accuracy of wrong adjunct island sentences from 0.69 (BASE) to 0.89. Smaller improvements are also observed for multiple other island effects when the model is trained with focal loss. Surprisingly, the model forced to predict the constituent labels does not perform well on island effects examples and the model trained with local attention outperforms the baseline by large margin on complex NP island and Wh island.

**Determiner Noun Agreement** Another notable change is within determiner noun agreement. This phenomenon tests whether a model recognizes incorrect noun after a determiner (e.g., *"that tables"* is unacceptable). The model trained with focal loss is better than the baseline model on multiple paradigms by large margins, especially on cases where adjective is inserted between the determiner and the noun. The accuracy of baseline model is improved from 81% to 88%. The second best modification is when the Transformer is trained with local attention, which consistently outperforms the baseline for all but two paradigms.

**Ellipsis and Irregular Forms** The model trained with local attention outperforms all other models on ellipsis, showing better ability to distinguish incorrectly omitted nouns (e.g. *"She took four heavy bags and he took five big"* has incorrectly omitted nouns at the end). Another consistent pattern arises in the irregular forms phenomenon, the model trained with auxiliary loss is better at recognizing incorrect past participle adjectives, suggesting the model assigns low probability to verbs when expecting a noun phrase, which could be a benefit from learning to predict the constituent labels.

**Negative Polarity Item** The last phenomenon we focus on is negative polarity items. We find that models trained with modified loss function outperform the baseline on identifying the correct scope of polarity item "ever" in the presence of the focus operator "only"(e.g., *"Those students who only Tim teaches ever pass the exam."* is incorrect as *ever* needs to be licensed by the word *only*, which should be in the main clause). The improvement is especially significant ($\sim$ 20 points) when evaluating the model trained with local attention. However, the baseline model is better at two other paradigms in the same phenomenon.

## 5 Related Work

Our work is closely related to recent analyses on the linguistic knowledge encoded within large pretrained LMs. One typical approach to probing the ingrained linguistic knowledge is through diagnostic classifiers, or probes (Alain and Bengio, 2017; Belinkov et al., 2017; Hewitt and Liang, 2019; Voita and Titov, 2020; Pimentel et al., 2020), a classifier trained with the intermediate representations of an LM. Previous works tend to evaluate the language models on set of multiple probing tasks (Liu et al., 2019; Conneau et al., 2018), each capturing a distinct linguistic phenomenon. Another type of probing relies on datasets constructed via linguistic rules that are specific to targeted linguistic phenomena (Jumelet and Hupkes, 2018; Marvin and Linzen, 2018; Warstadt et al., 2020b,a). Previous works have intervened at least two aspects of LM training: (1) the size of training data (van Schijndel et al., 2019; Zhang et al., 2021) and (2) the training task (Linzen et al., 2016; Ravfogel et al., 2019).

## 6 Conclusion

To complement recent analyses on the linguistic knowledge encoded by released Transformer LM checkpoints, we investigate four Transformer language models, each trained with slightly different settings. We evaluate these variants on BLiMP, a targeted evaluation set to probe the language models' capability of various linguistic phenomena. Our results show that although the averaged performance is similar after applying those changes, there are promising gains on local paradigms. We hope our work could shed light on future research into more effective learning of syntactic knowledge by Transformer language models.

## References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. 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.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. 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)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection.

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.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of RNNs with synthetic variations of natural languages. 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 3532–3542, Minneapolis, Minnesota. Association for Computational Linguistics.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.

Simeng Sun and Mohit Iyyer. 2021. Revisiting simple neural probabilistic language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5181–5188, Online. Association for Computational Linguistics.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. 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 5831–5837, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łũkasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020b. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Code-switching language modeling using syntax-aware multi-task learning. In *Proceedings of the Third Workshop* *on Computational Approaches to Linguistic Code-Switching*, pages 62–67, Melbourne, Australia. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? 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 1112–1125, Online. Association for Computational Linguistics.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Multi-task learning with language modeling for question generation. 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 3394–3399, Hong Kong, China. Association for Computational Linguistics.

# A Evaluation on BLIMP

| Phenomena | Paradigms | Base | FL | ML | AL | LA |
|---|---|---|---|---|---|---|
| Island Effects | adjunct_island | 0.69 | 0.81 | 0.89 | 0.85 | 0.69 |
| | complex_NP_island | 0.50 | 0.46 | 0.48 | 0.50 | 0.55 |
| | coordinate_structure_constraint_complex_left_branch | 0.42 | 0.39 | 0.38 | 0.33 | 0.33 |
| | coordinate_structure_constraint_object_extraction | 0.74 | 0.78 | 0.77 | 0.67 | 0.80 |
| | left_branch_island_echo_question | 0.48 | 0.49 | 0.46 | 0.42 | 0.40 |
| | left_branch_island_simple_question | 0.34 | 0.41 | 0.37 | 0.31 | 0.41 |
| | sentential_subject_island | 0.31 | 0.41 | 0.40 | 0.39 | 0.37 |
| | wh_island | 0.66 | 0.63 | 0.62 | 0.55 | 0.71 |
| Anaphor Agreement | anaphor_gender_agreement | 0.96 | 0.95 | 0.91 | 0.96 | 0.95 |
| | anaphor_number_agreement | 0.98 | 0.98 | 0.96 | 0.98 | 0.97 |
| Argument Structure | animate_subject_passive | 0.69 | 0.67 | 0.67 | 0.69 | 0.68 |
| | animate_subject_trans | 0.48 | 0.46 | 0.45 | 0.48 | 0.47 |
| | causative | 0.68 | 0.66 | 0.65 | 0.71 | 0.67 |
| | drop_argument | 0.52 | 0.49 | 0.51 | 0.48 | 0.51 |
| | inchoative | 0.64 | 0.64 | 0.64 | 0.62 | 0.66 |
| | intransitive | 0.57 | 0.57 | 0.58 | 0.58 | 0.57 |
| | passive_1 | 0.72 | 0.71 | 0.72 | 0.73 | 0.74 |
| | passive_2 | 0.72 | 0.72 | 0.71 | 0.72 | 0.70 |
| | transitive | 0.70 | 0.70 | 0.70 | 0.71 | 0.69 |
| Determiner Noun Agreement | determiner_noun_agreement_1 | 0.93 | 0.93 | 0.94 | 0.94 | 0.93 |
| | determiner_noun_agreement_2 | 0.94 | 0.95 | 0.94 | 0.93 | 0.95 |
| | determiner_noun_agreement_irregular_1 | 0.84 | 0.83 | 0.84 | 0.84 | 0.83 |
| | determiner_noun_agreement_irregular_2 | 0.87 | 0.88 | 0.86 | 0.86 | 0.87 |
| | determiner_noun_agreement_with_adj_2 | 0.81 | 0.88 | 0.84 | 0.84 | 0.86 |
| | determiner_noun_agreement_with_adjective_1 | 0.84 | 0.88 | 0.85 | 0.84 | 0.86 |
| | determiner_noun_agreement_with_adj_irregular_1 | 0.73 | 0.76 | 0.75 | 0.75 | 0.76 |
| | determiner_noun_agreement_with_adj_irregular_2 | 0.77 | 0.82 | 0.79 | 0.79 | 0.84 |
| Subject Verb Agreement | distractor_agreement_relational_noun | 0.85 | 0.82 | 0.81 | 0.86 | 0.82 |
| | distractor_agreement_relative_clause | 0.72 | 0.73 | 0.76 | 0.72 | 0.73 |
| | irregular_plural_subject_verb_agreement_1 | 0.83 | 0.82 | 0.82 | 0.82 | 0.82 |
| | irregular_plural_subject_verb_agreement_2 | 0.93 | 0.91 | 0.92 | 0.92 | 0.89 |
| | regular_plural_subject_verb_agreement_1 | 0.93 | 0.92 | 0.92 | 0.92 | 0.91 |
| | regular_plural_subject_verb_agreement_2 | 0.89 | 0.88 | 0.88 | 0.87 | 0.86 |
| Ellipsis | ellipsis_n_bar_1 | 0.70 | 0.73 | 0.75 | 0.69 | 0.78 |
| | ellipsis_n_bar_2 | 0.82 | 0.84 | 0.79 | 0.82 | 0.84 |
| Control & Raising | existential_there_object_raising | 0.75 | 0.72 | 0.69 | 0.72 | 0.68 |
| | existential_there_subject_raising | 0.82 | 0.85 | 0.81 | 0.81 | 0.84 |
| | expletive_it_object_raising | 0.74 | 0.78 | 0.69 | 0.75 | 0.70 |
| | tough_vs_raising_1 | 0.44 | 0.45 | 0.49 | 0.43 | 0.53 |
| | tough_vs_raising_2 | 0.87 | 0.91 | 0.86 | 0.89 | 0.86 |
| Quantifiers | existential_there_quantifiers_1 | 0.97 | 0.97 | 0.95 | 0.96 | 0.95 |
| | existential_there_quantifiers_2 | 0.16 | 0.24 | 0.12 | 0.16 | 0.16 |
| | superlative_quantifiers_1 | 0.89 | 0.85 | 0.86 | 0.71 | 0.92 |
| | superlative_quantifiers_2 | 0.80 | 0.71 | 0.78 | 0.74 | 0.80 |
| Irregular Forms | irregular_past_participle_adjectives | 0.88 | 0.93 | 0.91 | 0.94 | 0.90 |
| | irregular_past_participle_verbs | 0.94 | 0.93 | 0.93 | 0.95 | 0.93 |
| NPI | matrix_question_npi_licensor_present | 0.14 | 0.17 | 0.26 | 0.15 | 0.22 |
| | npi_present_1 | 0.58 | 0.53 | 0.54 | 0.47 | 0.59 |
| | npi_present_2 | 0.69 | 0.60 | 0.63 | 0.57 | 0.61 |
| | only_npi_licensor_present | 0.88 | 0.91 | 0.87 | 0.94 | 0.90 |
| | only_npi_scope | 0.66 | 0.79 | 0.82 | 0.77 | 0.84 |
| | sentential_negation_npi_licensor_present | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 |
| | sentential_negation_npi_scope | 0.50 | 0.60 | 0.56 | 0.49 | 0.58 |
| Binding | principle_A_case_1 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | principle_A_case_2 | 0.88 | 0.88 | 0.88 | 0.90 | 0.90 |
| | principle_A_c_command | 0.61 | 0.64 | 0.62 | 0.65 | 0.63 |
| | principle_A_domain_1 | 0.98 | 0.99 | 0.98 | 0.98 | 0.98 |
| | principle_A_domain_2 | 0.69 | 0.71 | 0.77 | 0.70 | 0.75 |
| | principle_A_domain_3 | 0.60 | 0.57 | 0.56 | 0.60 | 0.63 |
| | principle_A_reconstruction | 0.49 | 0.51 | 0.45 | 0.53 | 0.41 |
| Filler Gap | wh_questions_object_gap | 0.73 | 0.72 | 0.75 | 0.70 | 0.79 |
| | wh_questions_subject_gap | 0.88 | 0.87 | 0.89 | 0.84 | 0.91 |
| | wh_questions_subject_gap_long_distance | 0.92 | 0.88 | 0.84 | 0.87 | 0.88 |
| | wh_vs_that_no_gap | 0.94 | 0.95 | 0.94 | 0.95 | 0.96 |
| | wh_vs_that_no_gap_long_distance | 0.97 | 0.98 | 0.98 | 0.98 | 0.96 |
| | wh_vs_that_with_gap | 0.49 | 0.50 | 0.46 | 0.48 | 0.45 |
| | wh_vs_that_with_gap_long_distance | 0.17 | 0.15 | 0.18 | 0.15 | 0.20 |

Table 3: **Base** stands for baseline model, **FL** stands for the model trained with focal loss ($\gamma = 2$), **ML** stands for the model trained with masked loss, the threshold $t = 0.9$, **AL** stands for model trained with auxiliary loss, the auxiliary task is to predict corresponding constituent label, **LA** stands for the model trained with local attention. The values below the baseline accuracy is marked in orange, above in blue.