Lexical Generalization Improves with Larger Models and Longer Training

Elron Bandel[1,2] **Yoav Goldberg**[1,3] **Yanai Elazar**[4,3]
[1]Computer Science Department, Bar Ilan University
[2]IBM Research [3]Allen Institute for Artificial Intelligence
[4]Paul G. Allen School of Computer Science
Engineering, University of Washington
elron.bandel@gmail.com

Abstract

While fine-tuned language models perform well on many tasks, they were also shown to rely on superficial surface features such as lexical overlap. Excessive utilization of such heuristics can lead to failure on challenging inputs. We analyze the use of lexical overlap heuristics in natural language inference, paraphrase detection, and reading comprehension (using a novel contrastive dataset), and find that larger models are much less susceptible to adopting lexical overlap heuristics. We also find that longer training leads models to abandon lexical overlap heuristics. Finally, we provide evidence that the disparity between models size has its source in the pre-trained model.[1]

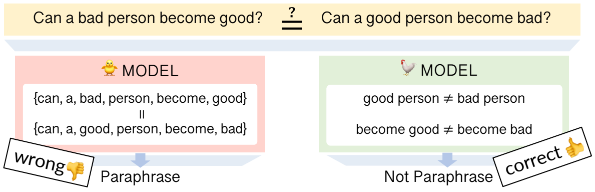

Can a bad person become good?
?=
Can a good person become bad?
🐣 MODEL
{can, a, bad, person, become, good}
=
{can, a, good, person, become, bad}
wrong👎
Paraphrase
🐓 MODEL
good person ≠ bad person
become good ≠ become bad
Not Paraphrase
correct👍

Figure 1: Illustration of the difference in the behavior of a too small, or briefly trained paraphrase detection 🐣MODEL, that relies on lexical overlap heuristic to make a (wrong) prediction. In contrast, 🐓MODEL is larger or is trained for longer, and does not rely on lexical overlap and predicts correctly.

1 Introduction

Pretrained Language Models (PLMs) dramatically improved the performances on a wide range of NLP tasks, resulting in benchmarks claiming to track progress of language understanding, like GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016) to be "solved". However, many works show these models to be brittle and generalize poorly to "out-of-distribution examples" (Naik et al., 2018; McCoy et al., 2019; Gardner et al., 2020).

One of the reasons for the poor generalization is models adopting superficial heuristics from the training data, such as, *lexical overlap*: simple match of words between two textual instances. For instance, a paraphrase-detection model that makes use of these heuristics may determine that two sentences are paraphrases of each other by simply comparing the bags-of-words of these sentences. While this heuristic sometimes works, it is also often wrong, as demonstrated in Figure 1. Indeed, while such heuristics are effective for solving in-domain large datasets, different tests expose that models often rely on these heuristics, and thus they are *right for the wrong reasons* (McCoy et al., 2019; Zhang et al., 2019). Since, different works tackled these problems, and propose different algorithms and specialized methods for reducing the use of such heuristics, and improve models' generalization (He et al., 2019; Utama et al., 2020; Moosavi et al., 2020; Tu et al., 2020; Liu et al., 2022).

In this work, we link the adoption of lexical overlap heuristic to size of PLMs and to the number of iterations in the fine-tuning process. We show that **a lot of the benefit from the above methods can be achieved simply by using larger PLMs and finetuning them for longer**.

We show that larger PLMs, and longer trained models are much less prone to rely on lexical overlap, **despite not being manifested on standard validation sets**. We validate these findings on three widely used PLMs: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019) and three tasks: Natural Language Inference (NLI), Paraphrase Detection (PD), and Reading Comprehension (RC). For RC, we collect a new contrastive test-set - ALSQA, based on SQuAD2.0 (Rajpurkar et al., 2018), while controlling on the lexical overlap between the texts and the questions, containing 365 examples.

[1] Code and data are available at: https://github.com/elronbandel/lexical-generalization.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4398–4410
December 7-11, 2022 ©2022 Association for Computational Linguistics

2 Related Work

This work relates to a line of research using behavioral methods for understanding model behavior (Ribeiro et al., 2020; Elazar et al., 2021a; Vig et al., 2020), and more specifically regarding the extent in which specific heuristics being used by models for prediction (Poliak et al., 2018; Naik et al., 2018; McCoy et al., 2019; Jacovi et al., 2021; Elazar et al., 2021b). The use of heuristics such as lexical overlap also typically point on the lexical and under sensitivity of models (Welbl et al., 2020), studied in different setups (Iyyer et al., 2018; Jia and Liang, 2017; Gan and Ng, 2019; Misra et al., 2020; Belinkov and Bisk, 2018; Ribeiro et al., 2020; Ebrahimi et al., 2018; Bandel et al., 2022).

Our work suggests that the size of PLMs affects their *inductive bias* (Haussler, 1988) towards the preferred strategy. Studies of inductive biases in NLP has gained attention recently (Dyer et al., 2016; Battaglia et al., 2018; Dhingra et al., 2018; Ravfogel et al., 2019; McCoy et al., 2020a). While Warstadt et al. (2020) studied the effect of the amount of pre-training data on linguistic generalization, we show that additional training iterations and larger models affects their inductive biases with respect to the use of lexical overlap heuristics.

Finally, Tu et al. (2020) show that the adoption of heuristics can be explained by the ability to generalize from a few examples that aren't aligned with the heuristics. They show that larger model size and longer fine-tuning can marginally increase the ability of the model to generalize from minority groups, an insight also discussed also by Djolonga et al. (2021). Our focus of lexical heuristics exclusively, together with robust fine-grained experimental setup concludes, unlike Tu et al. (2020), that the change in lexical heuristic adoption behavior is consistent and non marginal.

3 Measuring Reliance on Lexical Overlap

Definition: Reliance on Lexical Overlap We say that a model makes use of *lexical heuristics* if it relies on superficial lexical cues in the text by making use of their identity rather than the semantics of the text.

3.1 Experimental Design

We focus on tasks that involve some inference over text pairs. We estimate the usage of the lexical-overlap heuristic by training a model on a *standard training set* for the task, and then inspecting models' predictions on a *diagnostic set*, containing high lexical overlap instances,[2] where half of the instances are consistent with the heuristic and half are inconsistent (e.g. the first two rows of each dataset in Figure 2, respectively). Models that rely on the heuristic will perform well on the consistent group, and significantly worse on the inconsistent ones.

Metric We define the HEURistic score (HEUR) of a model on a diagnostic set as the difference in model performance on the consistent examples, and its performance on the inconsistent examples. Higher HEUR values indicate high use of the lexical overlap heuristic (bad) while low values indicate lower reliance on the heuristic (good).

3.2 Data

For the **training sets** we use the MNLI dataset (Williams et al., 2018) for Natural Language Inference (NLI; Dagan et al., 2005; Bowman et al., 2015), the Quora Question Pairs (QQP; Sharma et al., 2019) for paraphrasing, and SQuAD 2.0 (Rajpurkar et al., 2018) for Reading Comprehension (RC). The corresponding high-lexical overlap diagnostic sets are described below:

HANS (McCoy et al., 2019) is an NLI dataset, designed as a challenging test set to test for reliance on different heuristics. Models that predict entailment when based solely on one of those heuristics - will fail in half the examples where the heuristic does not hold. We focus on the lexical overlap heuristic part, termed HANS-Overlap.

PAWS (Zhang et al., 2019) is a paraphrase detection dataset containing paraphrase and non-paraphrase pairs with high lexical overlap. The pairs were generated by controlled word swapping and back translation, with a manually filtering validation step. We use the Quora Question Pairs (PAWS-QQP) part of the dataset.

ALSQA To test the lexical overlap heuristic utilization in Reading Comprehension models, we create a new test set: Analyzing Lexically Similar QA (ALSQA). We augment the SQuAD 2.0 dataset (Rajpurkar et al., 2018) by asking crowdworkers to generate questions with high context-overlap

[2] We define the *lexical overlap* between pair of texts as the number of unique words that appear in both texts divided by the number of unique words in the shorter text. The definition for ALSQA is similar, but we consider lemmas instead of words, and ignore function words and proper names.

Dataset	Text1	Text2	Label
HANS	The banker near the judge saw the actor. The doctors visited the lawyer.	The banker saw the actor. The lawyer visited the doctors.	E ✓ NE ✗
PAWS	What should I prefer study or job? Can a bad person become good ?	What should I prefer job or study? Can a good person become bad?	P ✓ NP ✗
ALSQA	..."downsize" revision of vehicle categories .By 1977 , GMś full - sized cars reflected the crisis. By 1979, virtually all " full - size " American cars had shrunk , featuring smaller engines and smaller outside dimensions. Chrysler ended production of their full - sized luxury sedans at the end of the 1981 model year ...	By which year did full sized American cars shrink to be smaller ? What vehicle category did Chrysler change to in 1977 ?	A ✓ NA ✗

Figure 2: Examples from the datasets. All examples have in-pair high lexical overlap. In ALSQA examples overlapping content words colored in blue . Text1 and Text2 correspond to the premise and hypothesis in HANS, the two sentences in PAWS and the context and question in ALSQA. The labels for HANS pairs are either Entailment (E) or Non-Entailment (NE). For PAWS the labels are Paraphrase (P) or Non-Paraphrase (NP). For ALSQA the labels are (A) Answerable (NA) Non-Answerable. If the lexical overlap point on the label it is marked as ✓ means it is *consistent-with-heuristic*.

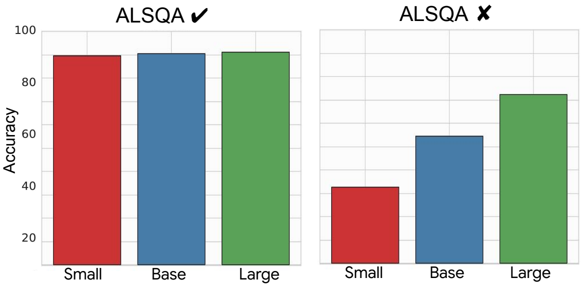


Figure 3: **"Larger is better"** Electra of different sizes perform equally on the subset of ALSQA that is consistent with the lexical overlap heuristic (✓), in the inconsistent subset (✗) larger models are less likely to adopt the heuristic, therefore, generalize better.

from questions with low overlap (These questions are paraphrases of the original questions). In the case of un-answerable questions, annotators were asked to re-write the question without changing its meaning and maintain the unanswerability reason.[3] ALSQA contains 365 questions pairs, 190 with answer and 174 without answer. Examples from the dataset are presented in Figure 2, and Appendix B.

4 Experiments and Results

Setup We experiment with 3 strong PLMs: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020), each with a few size variants. Since performance after finetuning can vary, both for in-domain (Dodge et al., 2020), and out-of-domain (McCoy et al., 2020b), we follow Clark et al. (2020) and finetune every model six times, with different seeds and learning rates (specified in Appendix C) and report the median of these results. All models achieve comparable results to those reported in the literature. For each model we consider different stages in the training process. The *early* variant is after one epoch, while the *late* variant is after six epochs.

The NLI and PD are text-pair classification tasks. However, RC is a span prediction task, as such, we explore two versions of this task: (1) the regular span prediction task, and (2) a text-pair classification task, where the goal is to predict if the questions is answerable or not from the text, which we call *Answerability*.

Finding I: Larger is Better (but it is not reflected on the dev set) We report the results in Table 1. Larger models perform consistently better on the lexical challenge across tasks and models: For NLI, the *early* BERT models gradually improve their HEUR score from 93.7 in the base version to 63.5 in the large version. Similarly for PD, the early RoBERTa improves the HEUR score from 84.4 to 76.0 from the base to large versions. The difference in the HEUR scores are remarkable, as the improvement on the standard validation set are relatively small with respect to the improvement on HEUR. For instance, the RoBERTa-early on HANS improve its dev performance from the base to large version by 2.2%, while HEUR improves relatively by 79.4%. Apart from the size trend, the absolute numbers between the tasks also differ significantly. RoBERTa-large practically "solved" this part of HANS, with a median accuracy of 96%. On the other hand, PAWS remains a much harder task, with the best

[3] Full details are provided in Appendix A.

Model-Stage	Size	Dev↑	✓↑	✗↑	HEUR↓	Δ
NLI						
BERT early	base	83.8	97.3	3.6	93.7	
	large	84.9	91.1	27.6	63.5	
BERT late	base	84.5	62.0	43.5	18.5	-75.2
	large	85.8	82.6	68.0	14.6	-48.9
RoBERTa early	base	85.9	99.4	9.7	89.7	
	large	87.8	99.8	81.4	18.4	
RoBERTa late	base	87.1	98.0	76.8	21.2	-68.5
	large	89.3	99.6	92.4	4.2	-14.2
Paraphrase Detection						
RoBERTa early	base	89.4	93.7	9.3	84.4	
	large	89.1	95.0	19.0	76.0	
RoBERTa late	base	91.6	90.1	21.9	68.2	-16.2
	large	91.9	94.8	23.9	70.9	-5.1
ELECTRA early	base	90.0	89.3	17.5	71.8	
	large	90.8	98.4	23.9	74.5	
ELECTRA late	base	91.8	89.5	35.1	54.4	-17.4
	large	92.5	93.7	42.0	51.7	-22.8
Question Answerability						
ELECTRA early	base	83.5	90.8	53.7	37.1	
	large	90.3	89.5	71.4	18.1	
ELECTRA late	base	83.5	90.5	54.6	35.9	-1.2
	large	91.0	91.1	72.3	18.8	0.7

Table 1: Results on all tasks considered in this study. We include both the dev-set results on the in-domain dataset (Dev), and the HEUR columns are reported on the high lexical overlap diagnostic sets. ✓ and ✗ refer to the subsets from the diagnostic sets that are consistent, and inconsistent with the heuristic, respectively. Δ refers to the difference in HEUR between the same model family of the same size, between the late and early stage of training (e.g. the difference between BERT-base in the late and early stage of training).

performing model is ELECTRA-large that obtains 56.6% accuracy.

Finding II: Longer is Better (but it is not reflected on the dev set) Next, we compare the *early* (after one iterations over the training data) and *late* (after six iterations) versions of models. To ease comparison, the Δ column indicates the difference between the HEUR score on the same model size, at the early and late stages of training. The differences show a constant improvement on this score from early to later versions of the trained models. Accordingly, the improvements on the standard development sets are again much smaller: for instance, on NLI using BERT-large the dev-score increases by 1% while the HEUR score improves relatively by 77% (see Figure 4)[4] This indicates that models tend to adopt the lexical overlap heuristic at early training stages, and abandon it later on.

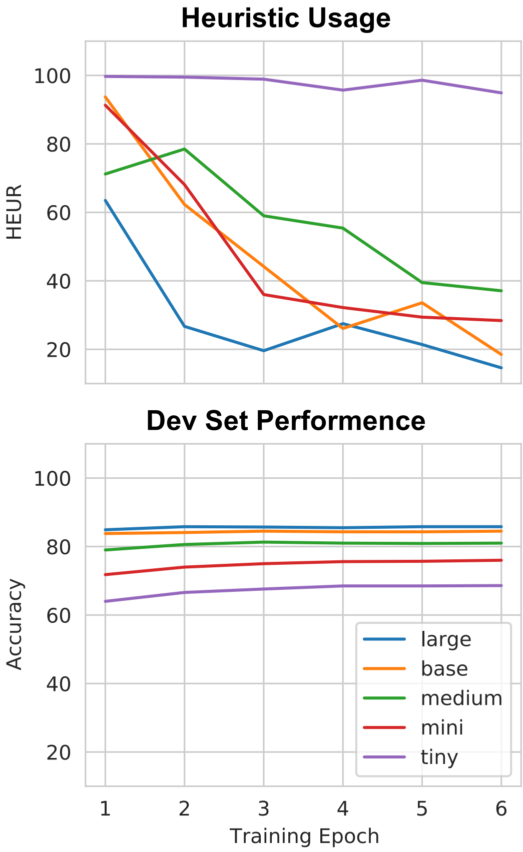


Figure 4: **"Longer is better"** Lexical Overlap HEUR of different sizes of BERT on HANS during training compared to dev set accuracy. While dev set results remain stagnant over training, the behavior of the model changes dramatically with regards to the reliance on the lexical heuristic, as reflected in the HEUR score.

Span-Prediction Objective We also compare models that were trained on the span-prediction task (RC) on SQuAD 2.0 to inspect if the task objective and loss affect the usage of lexical heuristic. We follow the same training setup solely for SQuAD 2.0 with the standard span-prediction task and report the results in Appendix F.2. We observe similar trends: larger or sufficiently fine-tuned models are less prone to use the overlap heuristic.

On the Source of Generalization Is the improvement in generalization based only on the capacity of the model being fine-tuned and the amount of fine-tuning, or are the larger pre-trained models themselves already better at generalization? Tu

[4] The trend seems to not hold in ELECTRA-large in Table 1 and both RoBERTa models in Table 8. However, we found that in an earlier training stage (middle of the first training iteration) the HEUR scores were significantly worse, while the dev scores were similar.

Model	Size	High Probability PPL	Δ	Low Probability PPL	Δ
BERT	base	2.5		11.2	
	large	2.4	-4.0%	9.5	-15.1%
RoBERTa	base	2.4		14.1	
	large	2.2	-8.3%	9.4	-33.3%

Table 2: The difference between the perplexity of pre-trained models sizes on probable and improbable sentences. On the probable sentences perplexity differences are marginal while on the improbable sentences differences are worse. Full results provided in Appendix F.

et al. (2020) show that larger models generalize better on challenging datasets by learning from a low probability sub-population of the training data (where the heuristics do not hold)[5]. However, it remains unclear if the capability to learn from improbable sub-populations, pre-exists from pre-training or emerges at fine-tuning. We speculate that **this capability has its source in the pre-trained model**.In the following experiment, we try and demonstrate that the larger pre-trained models are indeed better at generalization. We show that larger pre-trained models perform better on improbable sub-populations of the pre-training data, while having a similar performance as smaller models on the probable population. To approximate the distribution of the pre-training texts, we use an LM ensemble, and sort 3000 unseen sentences by their probability assigned by the ensemble. Doing so, we observe that larger LMs achieve considerably better perplexity on the improbable sentences, compared to the small models (15-33% relative improvement), while having comparable perplexity on the probable sentences (4-8% improvement). We provide more details and the full results in Appendix E. One possible explanation is that the larger models have higher memorization capacity (Tirumala et al., 2022) that can be helpful for learning from large variety of improbable sub-populations (Feldman, 2020). This pre-training skill can be utilized for generalizing from improbable groups, such as the examples where the heuristic does not hold, later in training[6].

5 Conclusions and Discussion

We show that (1) larger pre-trained models are less prone to make use of lexical overlap heuristics in fine-tuning; and (2) longer fine-tuning may reduce the use of such heuristics. These findings raise questions on the training dynamics of fine-tuned LMs (Saphra and Lopez, 2019; Chiang et al., 2020), which should be explored in future work: What kind of inductive biases does large models posses towards better generalization abilities? What makes them adopt and abandon overlap heuristics with minimal signal from the training data? We suggest that larger models are better at learning from improbable groups in the data due to their larger capacity. To support this claim we show that larger PLMs achieve better perplexity mainly in less probable sentences. Our work is the first to suggest that sufficiently-trained large PLMs are capable at arriving at solutions that are not wrongly reliant on lexical overlap. Such models should be used as baselines when developing techniques to alleviate the reliance of lexical heuristics (He et al., 2019; Utama et al., 2020; Moosavi et al., 2020) and assessing progress (Bowman, 2022).

6 Limitations

This work focuses solely on the effect of the lexical overlap heuristic and how models adopt and abandon it. As such, we study in depth this heuristic and its source of origin, but we cannot make broader claims on the quality of larger models and longer training, in general. More research is needed to study the effect of these models in other scenarios and heuristics to gain better understanding of their capabilities. Additionally, this work was mostly conducted two years ahead of publication, and while the size of the models and the length of training seemed to be enough back then, by today's standards they could be enlarged. Finally, like other empirical works in this field, this work makes broad claims over a large population (the possible space of all pre-trained language models) based on observation of a small sample from it. As with other works in this field, we do believe our conclusions to be useful despite the small-sample issue.

[5] Less then 0.1% of the training data (McCoy et al., 2019)

[6] Additionally, the link between the perplexity and the skill of learning from improbable data populations, might explain why models with better perplexity (such as RoBERTa in comparison to BERT of the same size, when PPL compared properly (Dudy and Bedrick, 2020)) tend to have lower HEUR scores (Table 4).

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No. 802774 (iEXTRACT). Yanai Elazar is grateful to have been supported by the PBC fellowship for outstanding PhD candidates in Data Science and the Google PhD fellowship for his PhD, where he spent most of his time on this project.

References

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 596–609, Dublin, Ireland. Association for Computational Linguistics.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Samuel Bowman. 2022. The dangers of underclaiming: Reasons for caution when reporting how NLP systems fail. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7484–7499, Dublin, Ireland. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proc. of EMNLP*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. Pretrained language model embryology: The birth of ALBERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

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.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 42–48, New Orleans, Louisiana. Association for Computational Linguistics.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander Nicholas D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lučić. 2021. On robustness and transferability of convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv*, abs/2002.06305.

Shiran Dudy and Steven Bedrick. 2020. Are some words worth more than others? In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 131–142, Online. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th*

Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021a. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021b. Back to square one: Bias detection, training and commonsense disentanglement in the winograd schema.

Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

David Haussler. 1988. Quantifying inductive bias: Ai learning algorithms and valiant's learning framework. *Artificial Intelligence*, 36(2):177–221.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020a. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020b. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Kanishka Misra, Allyson Ettinger, and Julia Rayz. 2020. Exploring BERT's sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635, Online. Association for Computational Linguistics.

N. Moosavi, M. Boer, Prasetya Ajie Utama, and Iryna Gurevych. 2020. Improving robustness by augmenting training sentences with predicate-argument structures. *ArXiv*, abs/2010.12510.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013, Online. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Lakshay Sharma, L. Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *ArXiv*, abs/1907.01041.

K. N. Bharadwaj Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *ArXiv*, abs/2205.10770.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

M K Vijaymeena and K Kavitha. 2016. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3:19–28.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages

217–235, Online. Association for Computational Linguistics.

Johannes Welbl, Pasquale Minervini, Max Bartolo, Pontus Stenetorp, and Sebastian Riedel. 2020. Undersensitivity in neural reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1152–1165, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Quentin Lhoest, Patrick von Platen, Yacine Jernite, Mariama Drame, Julien Plu, Julien Chaumond, Clement Delangue, Clara Ma, Abhishek Thakur, Suraj Patil, Joe Davison, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angie McMillan-Major, Simon Brandeis, Sylvain Gugger, François Lagunas, Lysandre Debut, Morgan Funtowicz, Anthony Moi, Sasha Rush, Philipp Schmidd, Pierric Cistac, Victor Muštar, Jeff Boudier, and Anna Tordjmann. 2020. Datasets. *GitHub*, 1.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

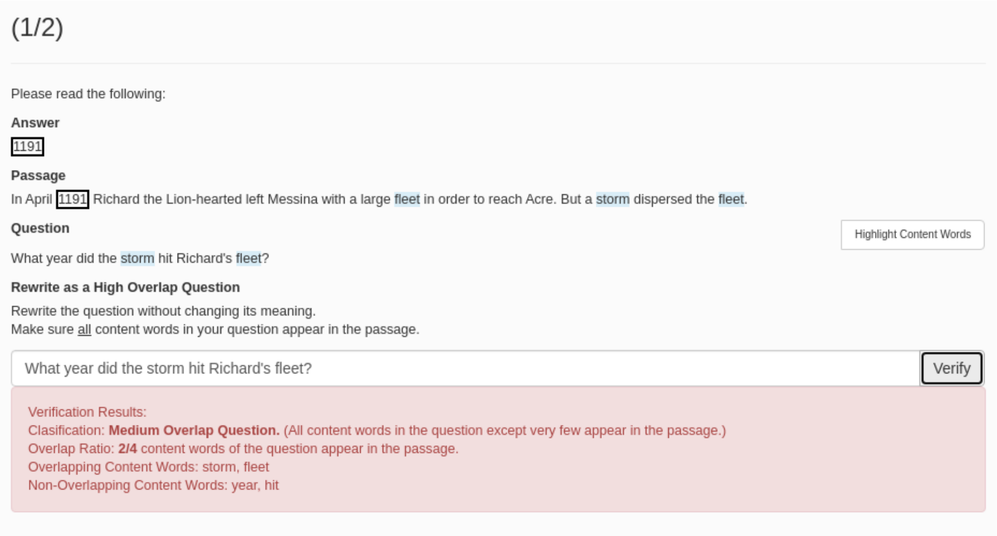
A Further Details on ALSQA

Lexical Overlap We consider three distinct classes of lexical overlap: High, Medium, and Low. We measure it as the overlap coefficient (Vijaymeena and Kavitha, 2016) between the bag of lematized[7] content words of two sentences. We define content words as words that are not named entities determiners or question words. We define lexical overlap coefficient higher than 0.85 as High, lower or equal to 0.85 and higher than 0.4 as Medium, and lower or equal than 0.4 as Low.

Collection Process To build the dataset, we sampled questions from the development set of SQuAD2.0 and calculated the lexical overlap level with their corresponding passages: High, Medium or Low. In unanswerable questions we considered the overlap between the question and the whole passage. In answerable questions we considered the lexical overlap between the question and the sentences containing the answer, in addition to one sentence before and after. We then filtered out high and medium overlap questions. Given a low overlap question, answer and a passage, the annotators were asked to rewrite the question such that (1) it will maintain its meaning in the context of the passage; (2) it will fit the High overlap class. We had an interactive lexical overlap validator to verify the paraphrased question indeed have high overlap. We also highlighted the overlapping and non-overlapping words in the question for facilitating the annotation procedure. In case the annotators did not manage to rewrite questions that satisfy the requirements they could provide an explanation and skip the question. Only less than 10% of the questions were skipped by annotators, and we removed these questions from the final dataset.

Annotators Preparation Following Roit et al. (2020); Pyatkin et al. (2020), we trained 8 annotators on a small amount of examples and gave them detailed feedback on their annotations. Then, we tested them on another small batch and proceeded with the dataset collection process only with annotators with high success rates on the test (5 annotators in total). Finally we added the annotations of an annotator to the final dataset only if we manually approved 95% of the annotations in 25% of the annotator work. We paid 0.4$ for each annotation including the ones we rejected.

[7] based on spaCy (Honnibal et al., 2020).


(1/2)

Please read the following:

Answer

1191

Passage

In April 1191 Richard the Lion-hearted left Messina with a large fleet in order to reach Acre. But a storm dispersed the fleet.

Question

What year did the storm hit Richard's fleet?

Rewrite as a High Overlap Question

Rewrite the question without changing its meaning.
Make sure all content words in your question appear in the passage.

What year did the storm hit Richard's fleet?

Verification Results:
Clasification: **Medium Overlap Question.** (All content words in the question except very few appear in the passage.)
Overlap Ratio: **2/4** content words of the question appear in the passage.
Overlapping Content Words: storm, fleet
Non-Overlapping Content Words: year, hit

Figure 5: A screenshot of the annotation interface used for collecting ALSQA. Verification process was employed to assist the annotators. The content words are identified both in the question and the passage, and are highlighted in blue after the annotator press the 'verify' button.

B Additional ALSQA Examples

In this section, we give a few more examples from the ALSQA dataset. Each example contains the original SQuAD2.0 context paragraph, a human generated question with high overlap with this paragraph, and the SQuAD2.0 answer for the question if it exists.

Example 1

Context

To the east is the Colorado Desert and the Colorado River at the border with Arizona, and the Mojave Desert at the border with the state of Nevada. To the south is the Mexico–United States border.

Question

What is at the south border with Arizona?

Answer

None

Example 2

Context

To the east is the Colorado Desert and the Colorado River at the border with Arizona, and the Mojave Desert at the border with the state of Nevada. To the south is the Mexico–United States border.

Question

What borders with the state of Nevada?

Answer

Mojave Desert

Example 3

Context

The San Bernardino-Riverside area maintains the business districts of Downtown San Bernardino, Hospitality Business/Financial Centre, University Town which are in San Bernardino and Downtown Riverside.

Question

What area maintains the districts University Town and business districts downtown?

Answer

Riverside

Example 4

Context

College sports are also popular in southern California. The UCLA Bruins and the USC Trojans both field teams in NCAA Division I in the Pac-12 Conference, and there is a longtime rivalry between the schools.

Question

The field teams from southern California were both from schools in sports in which NCAA group?

Answer

Division I

Example 5

Context

Even before the Norman Conquest of England, the Normans had come into contact with Wales. Edward the Confessor had set up the aforementioned Ralph as earl of Hereford and charged him with defending the Marches and warring with the Welsh. In these original ventures, the Normans failed to make any headway into Wales.

Question

Who made Edward the Confessor into an Earl?

Answer

None

Example 6

Context

The further decline of Byzantine state-of-affairs paved the road to a third attack in 1185, when a large Norman army invaded Dyrrachium, owing to the betrayal of high Byzantine officials. Some time later, Dyrrachium—one of the most important naval bases of the Adriatic—fell again to Byzantine hands.

Question

What one of the most important naval bases of the Adriatic-fell to the Normans?

Answer

None

C Training Details And Reproducibility

The finetuning was done using PyTorch finetuning code can be found on huggingface's transforemers (Wolf et al., 2020) github repository. For QQP and MNLI we used the following code: `https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue.py` and for SQuAD2.0: `https://github.com/huggingface/transformers/blob/main/examples/pytorch/question-answering/run_qa.py`. All experiments were run on 4 Nvidia GeForce RTX GPUs, and the training times varied from 10 GPU hours for the large models to 20 minutes for the tiny models. The hyper parameter selection procedure was done according to the finetuning protocol described in Liu et al. (2019). The hyper parameters[8] and huggingface's datasets paths are described in Table 3. The specifications of the different transformers model sizes mentioned in this paper, including their number of parameters are provided in Table 4.

D Datsets Details

The standard datasets for each task, as well as HANS, were taken from the huggingface's dataset library. PAWS-QQP was generated by the code in `https://github.com/google-research-datasets/paws`.

Sizes of the Different Datasets The details about the sizes of different datasets mentioned in this work can be found in Table 5.

E On the Source of Generalization: Elaboration

To approximate the distribution of the pre-training texts, we use a language models ensemble from different sizes. We use an ensemble to reduce the affect of the size of a single language model when computing sentence probabilities. In practice, we average the probabilities assigned by gpt2-large, gpt2-medium and gpt2-base. Using the probabilities assigned by the models ensemble we sort 3000 unseen sentences. The sampled sentences were not part of the models' training data, yet come from a similar distribution. In practice, we use the 1000 newest Wikipedia articles from the 15 June 2022 dump (the oldest article was created

[8] when using two learning rates the first learning rate used with seed 1-3 and the second 4-6.

checkpoint	dataset	scheduler	warmup pr	batch	lr	epochs	seeds
prajjwal1/bert-tiny	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
prajjwal1/bert-mini	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
prajjwal1/bert-medium	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
bert-base-uncased	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
bert-large-uncased	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
roberta-base	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
roberta-large	glue/mnli	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
roberta-base	glue/qqp	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
roberta-large	glue/qqp	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
electra-small	glue/qqp	linear	0.0	32	2e-5	6	[1,...,6]
electra-base	glue/qqp	linear	0.0	32	2e-5	6	[1,...,6]
electra-large	glue/qqp	linear	0.0	32	2e-5	6	[1,...,6]
roberta-base	squad_v2	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
roberta-large	squad_v2	linear	0.06	32	2e-5,3e-5	6	[1,...,6]
electra-small	squad_v2	linear	0.0	32	2e-5	6	[1,...,6]
electra-base	squad_v2	linear	0.0	32	2e-5	6	[1,...,6]
electra-large	squad_v2	linear	0.0	32	2e-5	6	[1,...,6]

Table 3: All the hyper-parameters used in this work for fine-tuning the different models. Checkpoint is the hugginface hub identifier of the model.

Model	# Layers	Hidden Dim	# Parameters
Tiny	2	128	4M
Mini	4	256	11M
Medium	8	512	41.M
Small	12	256	14M
Base	12	768	110M
Large	24	1024	355M

Table 4: Specifications of the different transformers model sizes mentioned in this paper, including their number of parameters.

Dataset	Train-Size	Dev-Size	✓
MNLI	392,702	9,815	
HANS-Overlap	10,000	10,000	50%
QQP	363,846	40,430	
PAWS-QQP	11,988	677	28.2%
SQuAD2.0	130,319	11,873	
ALSQA	-	365	50%

Table 5: Details about the sizes of different datasets mentioned in this work. The percentage of positive labeled examples in each testing dev-set presented under the column titled with '✓'.

		Probable		Improbable	
Model	**Size**	**PPL**	**Δ**	**PPL**	**Δ**
BERT	tiny	13.9		75.6	
	mini	5.6	-59.7%	31.7	58.0%
	med	3.2	-42.8%	16.0	49.5%
	base	2.5	-21.8%	11.2	30.0%
	large	2.4	-4.0%	9.5	15.1%
RoBERTa	base	2.4		14.1	
	large	2.2	-8.3%	9.4	33.3%
	rarity	21.3		161.9	

Table 6: The difference between the perplexity of pre-trained models on probable and improbable sentences. The relative improvement of every model size on top of its between the perplexity of every model It can be seen that while the differences in perplexity between sizes of models are more then 3 times larger in rare sentences.

in the 13st of June 2022, well ahead the model's training time). We then take the 20% of sentences with the highest probability assigned by the language model ensemble, and mark them as *probable* sentences and the lowest 20% as *improbable*. Then, we test the average perplexity that the different BERT and RoBERTa models assign to each group. Since these are MLM-based models, we compute the perplexity as defined by (Devlin et al., 2019), and later refered as pseudo perplexity (Salazar et al., 2020) and calculated as exponent of the pseudo log likelihood (Wang and Cho, 2019). For a sentence $W = w_1, ..., w_n$ and $P_{mlm}(W)_i$ (the probability for word i in a sentence to be w_i when unmasked by model P_{mlm}) the pseudo log likelihood is defined to be: $PLL(W) = \sum_{i=1}^{n} log(P_{mlm}(W)_i)$. The corresponding perplexity is $PPL(W) = exp(PLL(W))$

The full results for the experiment are presented in Table 6.

F Additional Results

In this section we describe additional results complementing the results described in the paper. The results support the claims made in the main paper by supplying evident for the existence of the same trends in more model size variants and settings.

F.1 Text Pair Classification

Additionally to results presented in the main paper, we also report the full HEUR results on the five BERT models we consider (large, base, medium, mini and tiny), across epochs on HANS in Figure 4. All models start with relativly high HEUR scores but during training HEUR scores gets better gradually. Additionally larger models tend to achive much better HEUR scores along training process. more results are presented in Table 7. Additionally the results of the models trained with span prediction objective show the same trend (Table 8) Notice that in some experiments the HEUR does not go down, for example RoBERTa in the span setings and ELECTRA large in the binary settings. If we look closely on those cases we can see that there is earlier stage where the HEUR was lower but the development set performance was not much lower. In ELECTRA large in the middle of the first iteration the development set results are 1.1 points lower than the early stage achieving high result of 89.2 and from this point to the end of training the HEUR goes down by 3.7 points. this might indicate that the models may adopt the heuristic at very early stage than abounded it later on.

F.2 Span Prediction Objective

In this section we supply additional details about the answerability accuracy results of the models trained with the standard span prediction training on SQuAD2.0. The results on both the dev set and ALSQA test set are provided in Table 8. We can see for example that ELECTRA-base has even higher dev set accuracy than ELECTRA-large, but the HEUR score of the larger model is much lower, indicating this model is less likely to use the lexical overlap heuristic. Comparison between the accuracy of ELECTRA early of different sizes on the consistent and inconsistent subsets of the data is visualized in Figure 3.

Model-Stage	Size	Dev↑	✓↑	✗↑	HEUR↓	Δ
			NLI			
BERT early	tiny	64.0	99.9	0.2	99.7	
	mini	71.8	95.1	3.8	91.3	
	med	79.0	84.6	13.4	71.2	
	base	83.8	97.3	3.6	93.7	
	large	84.9	91.1	27.6	63.5	
BERT late	tiny	68.6	97.6	2.7	94.9	-4.8
	mini	76.0	63.7	35.3	28.4	-42.8
	med	81.0	70.8	33.7	37.1	-54.2
	base	84.5	62.0	43.5	18.5	-75.2
	large	85.8	82.6	68.0	14.6	-48.9
RoBERTa early	base	85.9	99.4	9.7	89.7	
	large	87.8	99.8	81.4	18.4	
RoBERTa late	base	87.1	98.0	76.8	21.2	-68.5
	large	89.3	99.6	92.4	4.2	-14.2
			Paraphrase			
RoBERTa early	base	89.4	93.7	9.3	84.4	
	large	89.1	95.0	19.0	76.0	
RoBERTa late	base	91.6	90.1	21.9	68.2	-16.2
	large	91.9	94.8	23.9	70.9	-5.1
ELECTRA early	small	87.7	93.2	6.2	87.0	
	base	90.0	89.3	17.5	71.8	
	large	90.8	98.4	23.9	74.5	
ELECTRA late	small	90.1	91.6	10.6	81.0	-6.0
	base	91.8	89.5	35.1	54.4	-17.4
	large	92.5	93.7	42.0	51.7	-22.8
			Answerability			
ELECTRA early	small	72.3	89.5	31.4	58.1	
	base	83.5	90.8	53.7	37.1	
	large	90.3	89.5	71.4	18.1	
ELECTRA late	small	72.9	89.7	32.6	57.1	-1.0
	base	83.5	90.5	54.6	35.9	-1.2
	large	91.0	91.1	72.3	18.8	0.7

Table 7: Results on all tasks considered in this study. We include both the dev-set results on the in-domain dataset (Dev), and the HEUR columns are reported on the high lexical overlap diagnostic sets. ✓ refers to the subset that is consistent with the heuristic whereas ✗ refers to the subset that is inconsistent with it.

Model-Stage	Size	Dev↑	✓↑	✗↑	HEUR↓	Δ
ELECTRA early	small	70.5	86.6	44.0	42.0	
	base	82.7	92.3	55.7	36.6	
	large	89.3	91.0	70.6	20.4	
ELECTRA late	small	76.4	82.0	56.0	26.0	-16.0
	base	85.1	91.5	60.3	31.2	-5.4
	large	91.6	89.4	77.7	11.7	-8.7
RoBERTa early	base	84.8	86.3	67.4	18.9	
	large	88.8	86.6	75.4	11.2	
RoBERTa late	base	86.2	87.1	66.9	20.2	1.3
	large	90.2	88.7	74.9	13.8	2.6

Table 8: Answerability results on ALSQA for models trained on SQuAD2.0 in span based finetuning. See Table 1 for the columns descriptions.