Assessing Combinational Generalization of Language Models in Biased Scenarios

Yanbo Fang
Certik
yanbo.fang@certk.com

Zuohui Fu and **Xin Dong**
Rutgers University
{zuohui.fu,xd48}@rutgers.edu

Yongfeng Zhang
Rutgers University
yongfeng.zhang@rutgers.edu

Gerard de Melo
Hasso Plattner Institute / University of Potsdam
gdm@demelo.org

Abstract

In light of the prominence of Pre-trained Language Models (PLMs) across numerous downstream tasks, shedding light on what they learn is an important endeavor. Whereas previous work focuses on assessing in-domain knowledge, we evaluate the generalization ability in biased scenarios through component combinations where it could be easy for the PLMs to learn shortcuts from the training corpus. This would lead to poor performance on the testing corpus, which is combinationally reconstructed from the training components. The results show that PLMs are able to overcome such distribution shifts for specific tasks and with sufficient data. We further find that overfitting can lead the models to depend more on biases for prediction, thus hurting the combinational generalization ability of PLMs.

1 Introduction

Transformer-based (Vaswani et al., 2017) pre-trained Language Models (PLMs) have enabled substantial performance gains across numerous downstream tasks (Devlin et al., 2019; Brown et al., 2020). To evaluate PLMs, existing work largely follows the scheme of sampling training and test data from the same distribution. In reality, given the productivity of human language, humans are widely assumed to interpret new linguistic utterances based on some notion of compositionality (Chomsky, 2006; Baroni, 2020).

In this paper, we want to investigate to what extent models are prone to using biases for prediction and how this may affect their performance on unseen instances requiring combinational inference. This relates to their generalization ability, which is regarded as a key challenge in building human-like models (Bommasani et al., 2021). We propose a method to assess a PLM's generalization capacity in classification tasks that require combinational generalization to overcome biases in the training data. Specifically, we modify an original dataset by recombining components of training data points to form unseen test data. Figure 1 provides a brief illustration of the principle. Based on the training data, the model can easily classify the ID instances. However, since we introduce a scenario with special hidden biases, a PLM that only picks up such training data biases would fail on the COOD instances in the test set. To handle those correctly, the combinational inference is required, i.e., drawing conclusions based on smaller fragments of text observed during training. Finally, there are also truly challenging genuine OOD instances that are not easily combinationally inferrable. Details of the hidden bias scenario are given in the bottom part

Figure 1: Example of data induction for 3-way sentiment classification. The top shows two training instances. The below part shows three generated instances (**ID**: In-Distribution, **COOD**: combinational Out-Of-Distribution, **OOD**: Out-Of-Distribution). In the table, labels in red are the combinations used for training and ID, whereas black and blue labels are used in the test set. *Blue* ones (COOD) can be inferred from the training data, while *black* ones (OOD with Label 3 in Template) cannot be combinationally inferred because the training set does not include such template.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 392–397
November 20–23, 2022.

of Figure 1 as well as later on in Section 3. Overall, our results suggest that PLMs possess excellent generalization abilities and avoid succumbing to the risky form of bias introduced in the training data. However, the performance depends on the task and data size.

2 Related Work

Probing Pre-trained Models. Numerous studies attempt to shed light on how PLMs learn (Rogers et al., 2020). Beyond understanding linguistic structures and semantics (Hewitt and Manning, 2019; Tenney et al., 2019) as well as world knowledge (Li et al., 2021), some studies show that PLMs possess a strong generalization ability across similar tasks and in out-of-distribution detection (Hendrycks et al., 2020; Utama et al., 2020; Chen et al., 2021; Geng et al., 2022).

Most prior work assesses PLMs based on the setting of test and training data stemming from the same distribution. This yields insights on standard in-task or in-domain learning, while in our work we are interested in the type of more generalizable knowledge acquired from the in-task training data. This relates to the robustness of PLMs, as the model can only do well on our test data if it pays attention to all components of the data rather than falling prey to biases in the training data.

Combinational Generalization. The combinational generalization here refers to the model's ability to properly handle unseen data samples consisting of fragments observed during training, and regard combinational generalization as a part of compositional generalization. Some studies investigate the compositional features and inductive biases of neural net models for sequence-to-sequence and generation tasks (Liska et al., 2018; Lake and Baroni, 2018), mostly at the phrase level, while we consider encoder models for classification tasks and focus on compositional inference connecting entire sentences.

One similar work is R&R (Akyürek et al., 2021), which also constructs data from fragments of training data. The major difference is that they incorporate the constructed data into training, while we use it for COOD evaluation.

Prompt-based Tuning. Prompt tuning has been proposed to reduce the gap between pre-training and fine-tuning on downstream tasks (Brown et al., 2020; Scao and Rush, 2021). It often involves adding templates to the data and predicting label names at the position of the [MASK] token (Schick and Schütze, 2021b,a). Inspired by prompt engineering, our work also involves the use of template engineering. However, we do not invoke them to elicit a PLM's prior knowledge, but as a core part of the input semantics. Additionally, finding the best templates and label names is not our focus, so we have not investigated automated prompt identification techniques (Shin et al., 2020; Gao et al., 2021), but we demonstrate that our results are coherent across different templates and label names.

3 Approach

3.1 Data Induction

Given an original training dataset $\mathcal{D}$ consisting of (x, y) pairs, where x is a training instance and y is its corresponding label, we induce a new dataset $\hat{\mathcal{D}}$. The latter consists of $(\hat{x}, \hat{y})$ pairs, created by adding templates and generating new labels based on $\mathcal{D}$. Each $\hat{x} = x \oplus t$ is a combination of an x and a template t appended at the end of x. For a general template "*It is <label>*", the *<label>* is replaced by the task-specific label names, so there are $|Y|$ unique templates for each unique label y_t. For example, if the task is sentiment classification, there could be templates "*It is positive*", "*It is negative*", and possibly "*It is neutral*". If y_t in the template t is consistent with the original label y, the new label $\hat{y}$ is 0, otherwise, it is 1.

To evaluate a PLM's generalization ability for combinational generalization, a biased scenario is constructed based on $\hat{\mathcal{D}}$. The model could easily just learn shortcuts from the training data, without accounting for generalization. The training data excludes certain kinds of combinations of inputs and templates as shown in Fig. 1, so these combinations are unseen when fine-tuning the PLM, but it may still be possible for the model to infer them compositionally from parts of the training data. This is what we assess in this paper.

Example. To help illustrate how to construct the biased data, we take the task of Natural Language Inference (Williams et al., 2018) as an example. We select K data points x for each label as training data. For data x with label $y =$ ENTAILMENT or CONTRADICTION, we concatenate x with corresponding consistent templates and add label $\hat{y} = 0$ ($2K$ instances). If a model only observed these specific combinations, it would be prone to picking

up the bias and misunderstanding combinations of the same x with another t. For further K instances of x with $y = \text{NEUTRAL}$, we append two inconsistent templates to construct the $\hat{x}$. This yields another $2K$ data points with $\hat{y} = 1$ and leads to a balance between instances with labels 0 and 1 ($2K$ instances each) in the training dataset. The test set will then also ask for new combinations.

3.2 Main Results

3.3 Fine-tuning

Given a data instance $(\hat{x}, \hat{y})$ from $\hat{\mathcal{D}}$ such that $\hat{x} = x_i \oplus t_i$, we invoke the PLM to obtain a representation $\text{enc}_\theta = \text{Encoder}_\theta(x_i \oplus t_i)$, where θ are the model parameters. Next, a linear classifier $w \in \mathbb{R}^{d \times 2}$ where d is the representation size for [CLS] is trained by optimizing the objective:

$$\text{argmax}_\theta P(\hat{y}_i \mid w \cdot \text{enc}_\theta([\text{CLS}])) = \text{argmax}_\theta \frac{\exp(w_{\hat{y}_i} \text{enc}_\theta([\text{CLS}]))}{\sum_{\hat{y}' \in \{0,1\}} \exp(w_{\hat{y}'} \text{enc}_\theta([\text{CLS}]))},$$

where $\text{enc}_\theta([\text{CLS}])$ is the vector for [CLS], $w_{\hat{y}}$ denotes the softmax scores for $\hat{y} \in \{0, 1\}$.

4 Experiments

4.1 Experimental Setup

Training details. We consider the pre-trained versions of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). For efficiency, we disregard sentences of over 100 tokens[1]. The batch size is 32 for base models and 8 for large models. The learning rate is 1×10^{-5} for all models. The evaluation metric we use is accuracy.

Datasets. For our assessment, we rely on Yelp reviews (Zhang et al., 2015) and MNLI (Williams et al., 2018), which each have 3 labels.[2] The label inventories are listed in Table 1.

The main results are reported in Table 2 for K =2,000. The notation A/B refers to the original dataset A adapted with template B. If A and B are consistent, the label is 0. If not, the label is 1.

The primary observation is that COOD generalization succeeds on Yelp-reviews but does not work as well for MNLI. MNLI is intrinsically harder than Yelp sentiment classification, yet the ID accuracy is high for both MNLI and Yelp. We conjecture that COOD generalization can succeed when the model can straightforwardly infer the label from the semantics of fragments of the input. In addition, we determine that the size of training data may be a factor affecting results in the following analysis (details in Section 5.1).

Second, all PLMs achieve low scores for OOD prediction. This suggests that, as expected, PLMs can, in this case, only use their learned bias to make predictions on such OOD instances. But larger models, such as RoBERTa$_{\text{Large}}$, can achieve better COOD and OOD scores than smaller models over all tasks. This may indicate that larger models may have better generalization capacity.

5 Discussion

5.1 How Do Training Data and Parameter Count Affect the Model?

There is a consensus that more training data coupled with a larger parameter count tends to benefit models for ID tests. So it is worth investigating whether these factors can also contribute to combinational generalization.

Regarding the number of parameters, as Table 2 shows, bigger models obtain better results on COOD and ID data. This finding illustrates that powerful models fit the source domain better and may exhibit stronger combinational generalization.

As for data quantities, we evaluated RoBERTa$_{\text{Base}}$ and RoBERTa$_{\text{Large}}$ with different K and plotted the results in Figure 3. Our observation is that both the ID and COOD accuracy are proportional to K. Yet, compared with ID, the performance of COOD is more vulnerable to the data size, while the OOD results remain low. This demonstrates that the training size can influence a model's generalization, but we can also observe the performance gap between ID and COOD closes as K increases. Compared with Yelp, MNLI appears to be more challenging in terms of generalization.

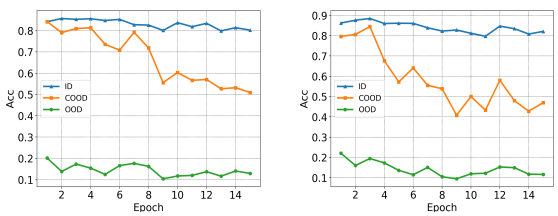


Figure 2: Results of BERT$_{\text{Base}}$ (left) and RoBERTa$_{\text{Base}}$ (right) on Yelp for K =3,000.

[1]This applies to the sum of the length of premise and hypothesis for MNLI.

[2]To achieve our combinational probing, note that the number of unique labels should be greater than 2.

Task name	Template	Label names
MNLI	⟨S⟩ It is ⟨LABEL⟩	entailment: entailment, neutral: neutral, contradiction: contradiction
Yelp	⟨S⟩ It is ⟨LABEL⟩	positive: great, neutral: okay, negative: terrible

Table 1: The default templates and label names in our experiments. ⟨S⟩ refers to original data.

Dataset	Models	ID				COOD		OOD		
		fst/fst	**sec/sec**	**neu/fst**	**neu/sec**	**fst/sec**	**sec/fst**	**fst/neu**	**sec/neu**	**neu/neu**
Yelp	1. $\text{BERT}_{\text{Base}}$	0.904	0.911	0.782	0.822	0.816	0.705	0.204	0.238	0.125
	2. $\text{BERT}_{\text{Large}}$	0.892	0.885	0.873	0.869	0.812	0.836	0.254	0.297	0.074
	3. $\text{RoBERTa}_{\text{Base}}$	0.913	0.871	0.838	0.784	0.773	0.754	0.247	0.244	0.054
	4. $\text{RoBERTa}_{\text{Large}}$	0.939	0.891	0.856	0.872	0.829	0.838	0.332	0.304	0.123
MNLI	5. $\text{BERT}_{\text{Base}}$	0.865	0.778	0.652	0.654	0.153	0.269	0.112	0.188	0.277
	6. $\text{BERT}_{\text{Large}}$	0.929	0.855	0.665	0.691	0.129	0.169	0.081	0.140	0.321
	7. $\text{RoBERTa}_{\text{Base}}$	0.921	0.857	0.756	0.786	0.356	0.239	0.145	0.133	0.314
	8. $\text{RoBERTa}_{\text{Large}}$	0.922	0.883	0.820	0.885	0.460	0.382	0.263	0.285	0.378

Table 2: Rows 1–4 report the main results on Yelp, while rows 5–8 provide results on MNLI. **fst**: positive/entailment, **neu**: neutral, **sec**: negative/contradiction.

Template	Label names	ID	COOD	OOD
Yelp (positive/neutral/negative)				
⟨S⟩ It is ⟨LABEL⟩	great/okay/terrible	0.855	0.761	0.173
⟨S⟩ It is ⟨LABEL⟩	cat/bird/dog	0.877	0.781	0.194
⟨S⟩ It is ⟨LABEL⟩	train/flight/car	0.881	0.763	0.182
⟨S⟩ It is ⟨LABEL⟩	terrible/great/okay	0.866	0.755	0.170
⟨S⟩ The sentence is ⟨LABEL⟩	great/okay/terrible	0.863	0.758	0.187
⟨S⟩ This sound like ⟨LABEL⟩	great/okay/terrible	0.850	0.764	0.185

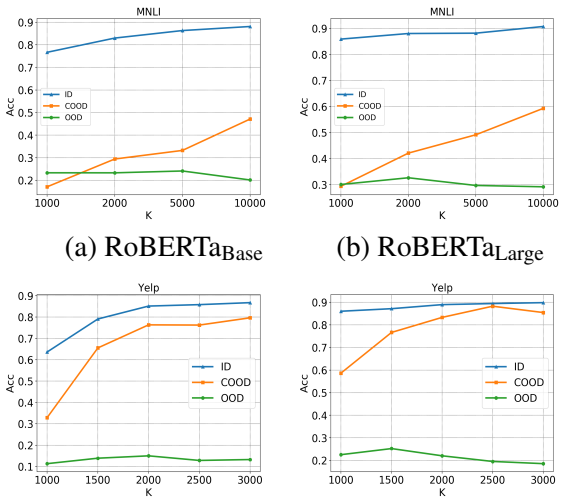
Table 3: $\text{RoBERTa}_{\text{Base}}$ Performance over Yelp dataset with different templates and label names. $K = 2000$. The order of label names denotes first and second known class and unknown class

5.2 Analysis of the Effect of Overfitting

Figure 2 depicts how the models perform on Yelp as the number of epochs increases and models increasingly overfit the data. As in Section 3.2, PLMs exhibit excellent performance on ID and COOD tasks and perform poorly on OOD tasks. ID and COOD accuracy both top out in nearly the same epoch, but as the number of epochs continues to increase, the results on COOD decrease more drastically than on ID. This suggests that when PLMs are overfitting, they tend to draw on biases and shortcuts for prediction. Another observation is that OOD accuracy may drop as well as ID and COOD as the number of epochs increase. We hypothesize that at early stages, the knowledge from pre-training still aids in prediction.

5.3 Effect of Label Names and Templates

We also compared the impact of different label names and templates. Based on the results shown in Table 3, the selection of label names and templates can affect the results. Even if the label names may not be intuitive, e.g., using label names DOG/CAT/BIRD or switching the order, models may obtain similar ID and COOD accuracy than in the original setting. This result can indicate that the prompt design may have a small impact on performance, as models can adjust to these differences.



(a) $\text{RoBERTa}_{\text{Base}}$ (b) $\text{RoBERTa}_{\text{Large}}$

(c) $\text{RoBERTa}_{\text{Base}}$ (d) $\text{RoBERTa}_{\text{Large}}$

Figure 3: Results for different training sizes K. The top two figures show results on MNLI, the bottom two show results on Yelp.

6 Conclusion

In this paper, we present a new method to probe the robustness of PLMs when subjected to biased data. Our findings include that (1) PLMs exhibit combinational generalization; (2) the combinational generalization is affected by the training data and parameter count; (3) overfitting is more harmful to a model's generalization ability than in-task ability.

Acknowledgments

We thank Tianyu Gao and all anonymous reviewers for their valuable feedback that help us improve this paper.

References

Ekin Akyürek, Afra Feyza Akyurek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. *ArXiv*, abs/2010.03706.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 375(1791):20190307–20190307. 31840578[pmid].

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, D. Card, Rodrigo Castellon, Niladri S. Chatterji, Annie Chen, Kathleen Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jackson K. Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258.

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.

Jiaao Chen, Dinghan Shen, Weizhu Chen, and Diyi Yang. 2021. HiddenCut: Simple data augmentation for natural language understanding with better generalizability. 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 4380–4390, Online. Association for Computational Linguistics.

Noam Chomsky. 2006. *Language and mind (3rd Ed.)*. Cambridge University Press.

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.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *Proceedings of the 16th ACM Conference on Recommender Systems*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL*.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *ACL/IJCNLP*.

Adam Liska, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional rnn in a haystack. *ArXiv*, abs/1802.06467.

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. *ArXiv*, abs/1907.11692.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *NAACL*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. *ArXiv*, abs/2009.07118.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *ArXiv*, abs/1905.05950.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

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. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.