DOI:10.1145/3596490

Shortcuts often hinder the robustness of large language models.

BY MENGNAN DU, FENGXIANG HE, NA ZOU, DACHENG TAO, AND XIA HU

Shortcut Learning of Large Language Models in Natural Language Understanding

NATURAL LANGUAGE UNDERSTANDING (NLU) is a subfield of artificial intelligence that requires computer software to comprehend input in the form of sentences. Representative NLU tasks include natural language inference (NLI), question answering (QA), and reading comprehension, among others. Furthermore, NLU has several real-world applications, including Alexa, Siri, and Google Assistant.

The major characteristic of NLU tasks is they are difficult and typically require world knowledge and commonsense reasoning. Recently, large language models (LLMs), such as BERT,[8] RoBERTa,[19] T5,[29] GPT-3,[4] have been reported to achieve state-of-the-art performance in a series of high-level NLU tasks. The LLM performance has reportedly been significantly higher than human performance. However, the superior performance has only been observed in the benchmark test data that has the same distribution as the training set. Recent studies indicate these LLMs are not robust and the models do not remain predictive when the distribution of inputs changes.[9,23,44] Specifically, these LLMs have low-generalization performance when applied to out-of-distribution (OOD) test data and are also vulnerable to various types of adversarial attack. This leaves us wondering: Why are these LLMs not robust? Have these LLMs mastered the high-level semantic understanding and reasoning we expect of them?

A major reason for the low robustness of LLMs is *shortcut learning*. The shortcut learning behavior has also been called other names in the literature, such as learning bias, superficial correlations, right for wrong reasons, and Clever Hans effect.[a] The shortcut learning behavior has been observed for a series of NLU tasks. For example, recent empirical analysis indicates the performance of BERT-like mod-

a The eponymous horse appeared to be capable of performing simple intellectual tasks, but relied on involuntary cues given by its handler.

» key insights

- **Shortcut learning is a major reason behind the lack of robustness in large language models (LLMs), which rely heavily on spurious correlations and non-generalized shortcuts in the training data rather than learning robust features for prediction.**
- **Shortcut learning is attributed to multiple factors, including biased training datasets, properties of the LLMs like model size, and the model training procedures like empirical risk minimization.**
- **Several methods have been proposed to mitigate shortcut learning, such as data debiasing, adversarial training, explanation regularization, and confidence regularization. However, existing methods have had limited success and there is a need for better understanding and solutions.**

IMAGE BY DIGITAL ABSTRACT ART

els for the NLI task could be mainly explained by relying on spurious statistical cues such as unigrams 'not,' 'do', 'is' and bigrams 'will not' (see Figure 1b).[12,23] Similarly, for the reading comprehension task, the models rely on the lexical matching of words between the question and the original passage, while ignoring the designed reading comprehension task.[16] The current standard approach to training LLM is to use empirical risk minimization (ERM) on NLU datasets that typically contain various types of artifacts and biases. As such, LLMs have learned to rely on dataset artifacts and biases and capture their spurious correlations with certain class labels as shortcuts for prediction. The shortcut learning behavior has significantly affected the robustness of LLMs (see Figure 1a), thus attracting increasing attention from the NLP community to address this problem.

In this work, we offer a comprehensive review of the shortcut learning problem in language models with a focus on medium-sized LLMs those typically having less than a billion parameters. The main emphasis is on the prevalent pre-training and fine-tuning paradigm utilized in NLU tasks. We cover the concept of shortcut learning and robustness challenges, detection approaches, characterization of the corresponding reasons, and mitigation approaches. We also provide a further discussion of future research directions and briefly discuss the challenges of shortcut learning posed by the prompt-based paradigm, especially regarding the massive language models which possess over a billion parameters, such as GPT-3 and T5.

Shortcut Learning Phenomena

Features captured by the model can be broadly categorized as useless features, robust features, and non-robust features (see Figure 2). Shortcut learning refers to the phenomenon that LLMs (especially those trained with standard ERM-based method) highly rely on non-robust features as shortcuts, failing to learn robust features and capture high-level semantic understanding and reasoning. Non-robust features do help generalization for development and test sets that share the same distribution with training data. However, they cannot generalize to OOD test sets and are vulnerable to adversarial attacks. Non-robust features are oriented from biases in the training data and come in different formats. Here, we introduce several representative ones.

▸ *Lexical bias:* Some lexical features have a high correlation of cooccurrence with certain class labels. These lexical features mainly consist of low-level functional words such as stop words, numbers, and negation words. A typical example is the NLI task, where LLMs are highly dependent on unintended lexical features to make predictions.[9,23] For example, these models tend to give contradiction predictions whenever negation words exist in the input samples, for example, 'never,' 'no.'

▸ *Overlap bias:* It occurs in NLU applications with two branches of text, for example, natural language inference, question answering, and reading comprehension. LLMs use the overlap of features between the two branches of inputs as spurious correlations as shortcuts. For example, reading comprehension models use the overlap between the passage and the question pair for prediction rather than solving the underlying task.[16] Similarly, question-answering models excel at test sets by relying on the heuristics of question and context overlap.[33]

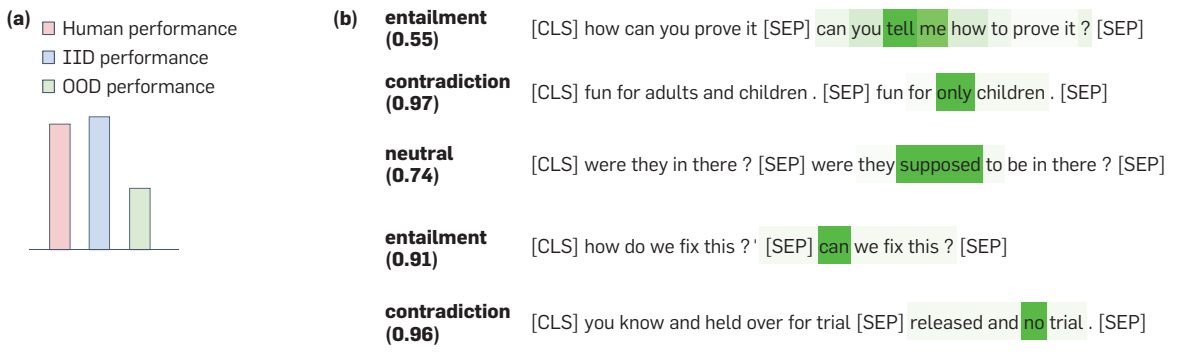
▸ *Position bias:* The distribution of the answer positions may be highly skewed in the training set for some applications. The LLMs would predict answers based on spurious positional cues. Take the question answering task for example, the answers lie only in the k^{th} sentence of each passage.[15] As a result, question answering models rely on this spurious cue when predicting answers.

▸ *Style bias:* The text style is a kind of pattern that is independent of semantics. Models have learned to rely on the erroneous text style as a shortcut rather than capturing the underlying semantics. Adversaries can use this style bias to launch adversarial attacks.[28]

Generalization and robustness challenge. The shortcut learning behavior could significantly hurt LLMs' *OOD generalization* as well as *adversar-*

Figure 1. Shortcut learning behavior and its negative impact, taking natural language inference (NLI) task for example.

The goal of NLI is to infer whether the relationship between two branches of input—premise and hypothesis—is entailment, contradiction, or neutral. (a) LLMs outperforms human performance for IID benchmark test set, while achieve much lower generalization performance on OOD test set. (b) A key reason is that LLMs primarily rely on the lexical bias and other kinds of shortcuts for prediction.



ial robustness. First, shortcut learning may result in significant performance degradation for OOD data. A common assumption is that training and test data are independently and identically distributed (IID). When LLMs are deployed in real-world applications with distribution shifts, this IID assumption will not hold any longer. This data typically does not contain the same types of bias and artifacts as the training data.

$$\begin{aligned} &\text{IID}: \boldsymbol{P}_{train}(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{P}_{test}(\boldsymbol{X}, \boldsymbol{Y}) \\ &\text{OOD}: \boldsymbol{P}_{train}(\boldsymbol{X}, \boldsymbol{Y}) \neq \boldsymbol{P}_{test}(\boldsymbol{X}, \boldsymbol{Y}) \end{aligned} \quad (1)$$

Using BERT-base as an example, there is a more than 20% reduction in accuracy on the OOD test set compared to the accuracy on the in-distribution test sets for NLU tasks.[10] To some extent, these models have solved the dataset rather than the underlying task. Second, shortcut learning produces models that are easily fooled by adversarial samples, which are generated when small and often imperceptible human-crafted perturbations are added to the normal input. One typical example is for the multiple-choice reading comprehension task.[37] BERT models are attacked by adding distracting information, resulting in a significant performance drop. Further analysis indicates these models are highly driven by superficial patterns, which inevitably leads to their adversarial vulnerability.

Shortcut Learning Detection

Here, we discuss methods to identify shortcut learning problems in NLU models.

Comprehensive performance testing. Traditional evaluations employ IID training-test split of data. The test sets are drawn from the same distribution as the training sets and thus share the same kind of biases as the training data. Models that simply rely on memorizing superficial patterns could perform acceptably on the IID test set. This type of evaluation has failed to identify the shortcut learning problem. Therefore, it is desirable to perform more comprehensive tests beyond the traditional IID testing.

First, the OOD generalization test has been proposed as an alternative to the IID test. Take the multi-genre natural language inference (MNLI) task for example, the HANS evaluation set is proposed to evaluate whether NLI models have syntactic heuristics: the lexical overlap heuristic, the subsequence heuristic, and the constituent heuristic.[20] Similarly, for the fact verification task, a symmetric test set is constructed that shares a philosophy like HANS.[32] These OOD tests have revealed dramatic performance degradation and exposed the shortcut learning problem of state-of-the-art LLMs.

Second, adversarial attacks could also be implemented to test the robustness of LLMs. For example, adversarial attacks have been used to reveal statistical bias in machine reading comprehension models.[16] Besides, the adversarial examples created through TextFooler[14] are used to test the generalization of commonsense reasoning models.[3] The results indicate the models have learned non-robust features and fail to generalize toward the main tasks associated with the datasets.

Third, randomization ablation methods are proposed to analyze whether LLMs have used these essential factors to achieve effective language understanding. For example, word order is a representative one among these significant factors. Recent ablation results indicate that word order does not matter for pretrained language models.[38] LLMs are pretrained first on sentences with randomly shuffled word order and then fine-tuned on various downstream tasks. The results show these models still achieve high accuracy. Similarly, another study[26] has observed that LLMs are insensitive to word order in a wide set of tasks, including the entire GLUE benchmark. These experiments indicate that LLMs have ignored the syntax when performing downstream tasks, and their success can almost be explained by their ability to model higher-order word co-occurrence statistics.

Explainability analysis. Model explainability is another effective tool the community has used to identify the shortcut learning problem. LLMs are usually considered black boxes, as their decision-making process is opaque and difficult for humans to understand. This presents challenges in identifying whether these models make decisions based on justified reasons or on superficial patterns. Explainability enables us to diagnose spurious patterns captured by LLMs.

The existing literature mainly employs the explanation in the format of feature attribution to analyze shortcut learning behavior in NLU models.[9] Feature attribution is the most

Figure 2. Features can be generally grouped into useless features, robust features, and non-robust features.

Non-robust features indicate various kinds of biases captured by the model, which are not robust in the OOD setting. In contrast, robust features denote features of high-level semantic understanding that are robust to changes in the input.

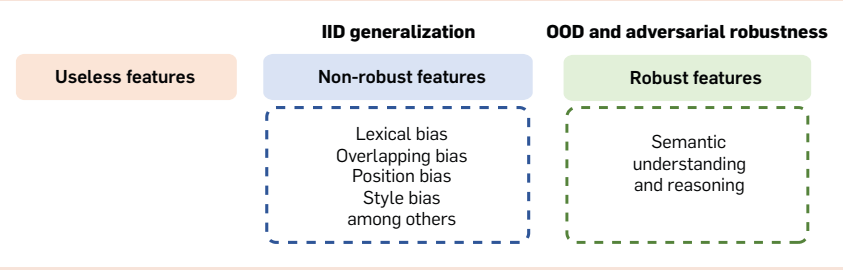
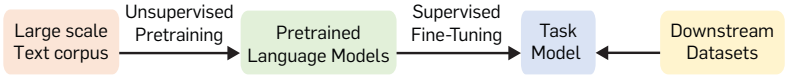


Figure 3. The pretraining then fine-tuning training paradigm.

Shortcut learning can be attributed to different factors in this pipeline, including pretrained language models, fine-tuning process, and downstream tasks.



representative paradigm among all explainability-based methods. For each token x_i within a specific input x, the feature attribution algorithm ψ will calculate the contribution score ψ_i, which denotes the contribution score of that token for model prediction. For example, the Integrated Gradient[41] interpretation method is used to analyze the model behavior of BERT-based models.[9] It is observed that LLMs rely on dataset artifacts and biases within the hypothesis sentence for prediction, including functional words and negation words, among others.[9] This shortcut learning behavior is summarized further using the long-tailed phenomenon. Specifically, the tokens in the training set could be modeled using a long-tailed distribution. The LLM models concentrate mainly on information on the head of the distribution, which typically corresponds to non-generalizable shortcut tokens. In contrast, the tail of the distribution is poorly learned, although it contains abundant information for an NLU task.

Beyond feature attribution, other types of explainability methods have also been used to analyze shortcut learning behaviors. For example, instance attribution methods have been used to explain model prediction by identifying influential training data, which can be used to explain decision making logic for the current sample of interest.[13] Empirical analysis indicates the most influential training data share similar artifacts, for example, high overlap between the premise and hypothesis for the NLI task. Furthermore, hybrid methods that combine feature attribution and instance attribution have also been used to identify artifacts in the data.[25] The resulting explanations have provided a more comprehensive perspective on the shortcut learning behavior of LLMs.

Origins of Shortcut Learning

The problem of learning shortcuts in LLM models for NLU tasks is a result of multiple factors present in the training pipeline (see Figure 3). In this section, we will delve into these reasons and give particular emphasis to three key elements: the training datasets, the LLM model, and the fine-tuning training procedure.

The problem of learning shortcuts in LLM models for natural language understanding tasks is a result of multiple factors present in the training pipeline.

Skewed training dataset. From a data standpoint, the NLU models' shortcut learning can be traced back to the annotation and collection artifacts of the training data. Here, the training data includes both the pre-training datasets as well as the downstream datasets (see Figure 3). Training sets are typically built through the crowd-sourcing process, which has the advantage of being low-cost and scalable. However, the crowd-sourcing process results in collection artifacts, where the training data is imbalanced with respect to features and class labels. Furthermore, when crowd workers author parts of the samples, they produce certain patterns of artifacts, that is, annotation artifacts.[12] Taking NLI as an example, the average sentence length of the hypothesis branch is shorter for the entailment category compared to the neutral category.[12] This suggests that crowd workers tend to remove words from the premise to create a hypothesis for the entailment category, leading to the overlap bias in the training data. Models trained on the skewed datasets will capture these artifacts and even amplify them during inference time.

LLMs models. The robustness of NLU models is highly relevant to the pre-fine-tuned language models. There are two key factors: model sizes (measured by the number of parameters) and pre-training objectives.

First, models with the same kind of architectures and pre-training objective but with different sizes could have significantly different generalization ability. It is shown that increasing the size of the model could lead to an increase in the representation power and generalization ability. From the empirical perspective, comparisons have been made between LLMs of different sizes but with the same architecture, for example, BERT-base with BERT-large, RoBERTa-base with RoBERTa-large.[2,43] The results show the larger versions generalize consistently better than the base versions, with a relatively smaller accuracy gap between the OOD and IID test data. Smaller models have fewer parameters than larger models and have a smaller model capacity. Therefore, smaller models are more prone to capture spurious patterns and are more dependent on data artifacts for

prediction. Another work[10] studies the impact of model compression on the generalizability of LLMs and finds that compressed LLMs are significantly less robust compared to their uncompressed counterparts. Compressed models with knowledge distillation have also been shown to be more vulnerable to adversarial attacks. From a theoretical perspective, a recent analysis supports there is a trade-off between the size of a model and its robustness, where large models tend to be more robust than smaller ones.[5]

Second, LLMs with similar model sizes but with different pretraining objectives also differ in the generalization ability. Here, we consider three kinds of LLMs: BERT, ELECTRA, and RoBERTa. BERT is trained with masked language modeling and next-sentence prediction. RoBERTa removes the next-sentence prediction from BERT and uses dynamic masking. ELECTRA is trained to distinguish between real input tokens and fake input tokens generated by another network. Empirical analysis shows these three models have significantly different levels of robustness.[27] For the Adversarial NLI (ANLI) dataset, it is shown that ELECTRA and RoBERTa have significantly better performance than BERT, for both the base and the large versions. Similarly, another study has shown that RoBERTa-base outperforms BERT-base around 20% in terms of accuracy on the HANS test set.[2] Because different architectures have distinct object functions during the pre-training stage, different inductive biases may be encoded by the models. This could possibly explain their differences in generalizability.

Model fine-tuning process. The learning dynamics could reveal what knowledge has been learned during model training. There are some observations. First, standard training procedures have a bias toward learning simple features, which we can refer to as the simplicity bias. The models are based mainly on the simplest features and remain invariant to complex predictive features. Moreover, it has been observed that the models give overconfident predictions for easy samples and low-confidence predictions for hard samples. Second, models tend to learn non-robust and easy-to-learn features at the early stage of training. For example, reading comprehension models have learned the shortcut in the first few training iterations, which has influenced further exploration of the models for more robust features.[16] Third, it has been experimentally validated that longer fine-tuning could lead to better generalization. Specifically, a larger number of training epochs will dramatically improve the generalizability of LLMs in NLU tasks.[43] The preference for non-robust features can be explained from the following perspective: The present LLM training methods can be considered as data-driven, corpus-based, statistical, and machine-learning approaches. It is postulated that while this data-driven paradigm may prove effective in certain NLP tasks, it falls short in relevance to the challenging NLU tasks that necessitate a deeper understanding of natural language.

Mitigation of Shortcut Learning

Here, we introduce approaches that alleviate the problem of shortcut learning. The goal is to improve OOD generalization and adversarial robustness while still exhibiting good predictive performance in IID datasets. These methods are motivated mainly by the insights obtained in the last section.

Data-centric mitigation approaches. *Dataset refinement* falls into the pre-processing mitigation family, with the aim of alleviating biases in the training datasets. First, when constructing new datasets, crowd workers will receive additional instructions to discourage the use of words that are highly indicative of annotation artifacts. Second, debiased datasets can also be developed by filtering out bias in existing data. For example, adversarial filtering is used to build a large-scale dataset for the NLI task to reduce annotation artifacts that can be easily detected by a committee of strong baseline methods.[51] As a result, models trained on this dataset must learn more generalizable features and rely on common sense reasoning to succeed. Third, we can also reorganize the train and test split, so the bias distribution in the test set is different from that in the training set. Lastly, various types of data augmentation methods have been proposed. Representative examples include counterfactual data augmentation, mix-up data augmentation, and syntactically informative example augmentation by applying syntactic transformations to sentences.

However, a drawback of this approach is that refining the dataset can only mitigate a limited number of recognized biases. The refined training set may not be completely free of biases and may still encompass statistical biases that are challenging for humans to identify. Thus, this could still negatively impact the model's performance.

Training samples reweighting. The main idea of reweighting is to place higher training weights on hard training samples, and vice versa.[32,44] It is also called worst-group loss minimization in some literature. The underlying assumption is that improving the performance of the worst group (hard samples) is beneficial to the robustness of the model. It is typically achieved through two-stage training. In the first stage, the weight indexing model is trained; and in the second stage, the predictions of the indexing model are used as weights to adjust the importance of a training instance. Both soft weights[44] and hard weights could be used in the second stage. Another representative example is focal loss, which is based on a regularizer to assign higher weights to hard samples that have less confident predictions.

Partitioning data into environments. This line of methods follows the principle of invariant risk minimization,[1] which encourages models to learn invariants in multiple environments. For example, training data has been partitioned into several non-IID subsets (that is, training environments), where spurious correlations vary across environments and reliable ones remain stable across environments.[42] The training scheme is designed to encourage the model to rely on stable correlations and suppress spurious correlations. Another work proposes an inter-environment matching objective by maximizing the inner product between gradients from different environments, with the goal of increasing model generalization.[35]

Model-centric mitigation methods, also known as named *robust learning* methods, typically augment the traditional ERM-based training paradigm with different degrees of prior

knowledge, explicitly or implicitly suppressing the model from capturing non-robust features. Some mitigation methods require the shortcuts be known a priori, while others assume the shortcuts are unknown.

Adversarial training aims to learn better representations that do not contain information about artifacts or bias in the data. It is typically implemented in two ways in the NLP domain:[30,40] First, the task classifier and adversarial classifier jointly share the same encoder.[40] The goal of the adversarial classifier is to provide the correct predictions for the artifacts in the training data. Then the encoder and task classifier can be trained to optimize the task objective while reducing the performance of the adversarial classifier in predicting artifacts. Second, adversarial examples are generated to maximize a loss function, and the model is trained to minimize the loss function. For example, the generator based on the masked language model is used to perturb the text to generate adversarial samples.[30] Despite the difference, both methods leverage the MinMax formulation during the debiasing process.

Explanation regularization. This category aims to regularize model training using prior knowledge established by humans.[18] Specifically, it is achieved by regularizing the feature attribution explanations with rationale annotations created by domain experts, to enforce the model to make the right predictions for the right reasons.[18] These systems are trained to explicitly encourage the network to focus on features in the input that humans have annotated as important and suppress the models' attention to superficial patterns. For the NLI task, natural language explanations have been used to supervise the models, to encourage the model to pay more attention to the words present in the explanations.[39] It has significantly improved the models' OOD generalization performance. Note that this type of method can only be used when prior knowledge is known in advance about shortcuts.

Product-of-expert (PoE). The goal is to train a debiased model by training it as an ensemble with a bias-only model.[7] This paradigm usually contains two stages. In the first stage, a bias-only model is explicitly trained to capture the bias of the dataset, for example, the hypothesis-only bias for the NLI task. During the second stage, the debiased model will be trained using cross-entropy loss, by combining its output with the output of the bias-only model: $\hat{p}_i = \text{softmax}(\log(p_i) + \log(b_i))$. The parameters of the bias-only model is fixed during this stage, and only the debiased model parameters are updated by backpropagation. The goal is to encourage the debiased model to utilize orthogonal information with information from the bias-only model to make predictions.

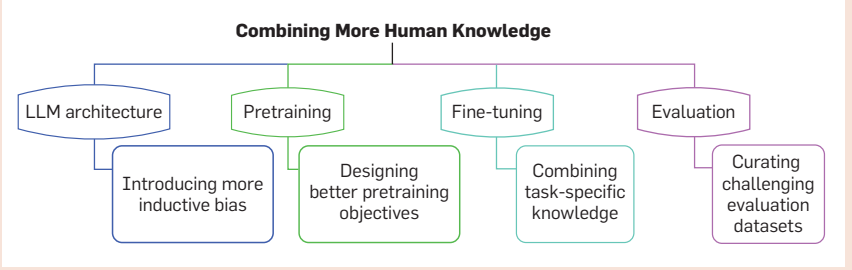
Confidence regularization. This mitigation scheme regularizes confidence in the model output, with the aim of encouraging the debiased model to give a higher uncertainty (lower confidence) for these biased samples. It is based on the observation that models tend to make overconfident predictions on biased examples. This relies on the training of a bias-only model to quantify the degree of bias of each training sample. The debiasing process is typically achieved through the knowledge distillation framework. In the first stage, the biased teacher model is trained using standard ERM loss, and the bias degree obtained from the bias-only model will be used to rescale the output distribution of the teacher model. In the second stage, the smoothed confidence values of the teacher model can be used to guide the training of the debiased model.[9]

Contrastive learning. Contrastive learning can be used to guide the training of representations. The goal is to construct the instance discrimination task to guide the model to capture the robust and predictive features, while suppressing the undesirable non-robust features. The instance discrimination task should be carefully designed, or it is possible to suppress robust predictive features.[31] A representative work presents a framework for mitigating spurious correlations using contrastive learning.[6] The method synthesizes a pair of factual and counterfactual inputs from the original text by masking identified causal and non-causal terms respectively. The model learns to associate the causal term with task labels by comparing the original text with its counterfactual counterpart, while learning to ignore non-causal features by contrasting with the factual pair. The framework leads to models that are less dependent on non-robust features and exhibit improved generalization performance.

IID and robustness trade-off? Another open question is about the connection between IID performance and OOD robustness performance. To the best of our knowledge, there are no consistent observations. For example, there is a linear correlation between IID performance and OOD generalization for different types of models introduced previously. On the contrary, most robust learning methods will sacrifice IID performance, although some of them could preserve IID performance. It deserves further research on the conditions under which the trade-off would occur. These insights could help the research community design robust learning frameworks that can simultaneously improve OOD and IID performance.

Figure 4. Combining more human knowledge to different stages of the pipeline.

Specifically, knowledge can be combined to the architecture of model, model training process (both pretraining and fine-tuning), and model evaluation process.



Future Research Directions

Despite the progress described in this article, there are still numerous research challenges. Here, we discuss po-

tential research directions that could be pursued by the community.

Introducing more domain knowledge. The current standard of LLM training is data-driven. This is problematic because the resulting models essentially perform low-level pattern recognition. It may be useful for low-level NLP tasks like named-entity recognition (NER), but it is nearly impossible to tackle the more difficult natural language understanding tasks. As a result, it is preferable to combine the data-driven scheme with domain knowledge by incorporating knowledge at various stages of training. Furthermore, more knowledge should be applied to the design of the model architecture and the model evaluation (see Figure 4).

Inductive bias to LLMs models. It is suggested to introduce more inductive bias into the model architecture to improve robustness and generalization beyond IID benchmark datasets. Recently, some work has begun to induce certain kinds of linguistic structure in neural architectures. For example, TableFormer is proposed for robust table understanding.[50] It proposes a structurally aware table-text encoding architecture, where tabular structural biases are incorporated through learnable attention biases. Although introducing linguistic-oriented biases to the model architectures might not result in the best performance for benchmark datasets, it is essential to improve generalization beyond IID benchmarks. Note that inductive biases are highly task-dependent and should be carefully designed for each specific task to accommodate its unique characteristic.

Better pre-training objectives. The pre-training objective also plays a crucial role in determining the OOD robustness of fine-tuned language models. As an example, recent studies have shown that pretrained BERT embeddings suffer from strong anisotropy, meaning the average cosine similarity is significantly higher than zero and word vectors cluster in narrow cones in the vector space.[11,17] This leads to word representations having a high similarity to unrelated words, impacting their expressive power and accuracy in downstream tasks. It is desirable to invest more effort in designing better pre-training objectives to improve model robustness. Recent studies indicate that choosing a better pretrained model could bring much better generalization performance than robust learning methods discussed earlier. For example, RoBERTa-base with a standard fine-tuning loss could even outperform the BERT-base with robust learning objectives in terms of generalization performance on the HANS test set.[2] This highlights the importance of pretraining in NLU model generalization performance and calls for increased community efforts to improve pretrained language models.

Adversarial training aims to learn better representations that do not contain information about artifacts or bias in the data.

Better fine-tuning approaches. NLU tasks may contain various types of bias, which are not fully known even by domain experts. This is distinct from the literature that works with the toy task (for example, Colored MNIST[1]), which typically contains a single type of bias and the bias is fully known. As a result, most existing mitigation methods for NLU tasks rely on human prior knowledge heuristics. Some examples include: weak models are more prone to capture biases, and non-robust models tend to give overconfident predictions for easy samples, among others. Unfortunately, this prior knowledge can only identify a limited number of biases in the data. Although it is possible to reduce the use of some identified shortcuts, models may still use other shortcuts for prediction. This could explain why existing mitigation methods only provide a limited improvement in generalization. As a result, it is suggested to incorporate more human-like common sense knowledge into the model training.

Curating challenging evaluation datasets. It is encouraging to see that some benchmark datasets for adversarial and OOD robustness have emerged. For example, adversarial GLUE is proposed for adversarial robustness evaluation, which contains 14 adversarial attack methods.[47] Despite these recent advances, it is necessary to continue curating difficult evaluation datasets that cover a wider range of NLU tasks, such as reading comprehension, and that cover a wider range of biases, such as those listed previously.

Revisiting the mitigation approaches. Existing mitigation methods have typically had limited mitigation performance. For example, for the MNLI task,

the accuracy for mitigated models with BERT-base as the backbone is consistently lower than 70% for the HANS test set.[44] Note that HANS is a balanced binary test set, where 50% is the accuracy of the random guess. The improvement in performance falls far short of our expectations. This brings up the following questions: What have the mitigation algorithms accomplished? How can mitigation performance be improved further? Debiased algorithms are thought to achieve better generalization because they can learn more robust features than biased models that rely primarily on non-robust features. However, this is not always the case with debiased algorithms. A recent work uses explainability as a debugging tool to analyze debiased models.[21] The analysis indicates the debiased models encode more biases in their inner representations. It is speculated the improved performance on the OOD data comes from the refined classification head. More research is needed to investigate whether the debiased model has captured more robust features and what is the source of their improved generalization. This also suggests an interesting research direction by only updating the biased classification head, as updating the entire model is typically difficult and time consuming.

In-depth theoretical understanding. In addition to the current empirical research, there is also a growing trend of preliminary theoretical research aimed at uncovering the shortcut learning behavior of DNN models.[22,34,46] For instance, using one-hidden-layer neural networks as the base model, one theoretical work uncovers that neural networks tend to exclusively rely on simplest and non-robust features, while remain invariant to other useful but more complex features.[34] This type of simplicity bias is one of the primary causes of low OOD generalization and adversarial vulnerability. Another theoretical study has investigated the reason behind superficial correlations from the optimization perspective.[46] By using a depth-2 ReLU network as an example, the study proposed the Gradient Starvation phenomenon, which states the gradient descent optimization methods tend to learn non-robust networks while slowing down the learning of robust and task-relevant features. Although these existing works provide insights into the reason of shortcut learning of shallow neural networks, there is still a lack of a solid theoretical understanding of why LLMs learn shortcuts. Further research is needed to fully explain this tendency in the context of LLMs.

Debiased algorithms are thought to achieve better generalization because they can learn more robust features than biased models that rely primarily on non-robust features. However, that is not always the case with debiased algorithms.

Taking inspiration from other directions. In addition, we can take inspiration from other relevant directions to address the shortcut learning issue of LLMs.

Domain adaptation and generalization. The robust learning approaches we discussed are closely relevant to domain adaptation and domain generalization. The three directions share the similarity that training and test sets are not from the same distribution, that is, there is a certain distribution shift. However, the objective of robust learning is distinct from domain adaptation, which aims to generalize to a specific target domain. In contrast, robust learning is closer to domain generalization, where both areas have the goal of generalizing over a range of unknown conditions. The NLP community can leverage the findings from the domain generalization area to design more robust learning methods for LLMs.

Long-tailed classification. Long-tailed classification addresses the issue of long-tailed distributed data, in which the head class has many training samples while the tail class has few. Shortcut learning can be treated as a special case of long-tailed classification, where easy samples correspond to the head class and hard samples represent the tail class. Some of the robust learning solutions (for example, reweighting) share a similar philosophy with approaches to the long-tailed classification problem. Leveraging ideas from approaches to long-tailed classification could improve the robustness of LLMs even further.

Algorithmic discrimination. Shortcut learning could also lead to discrimination and unfairness in deep learning models. In contrast to the general bias captured by the models, the spurious patterns here usually correspond to societal biases in terms of humans (for example, racial bias and gender bias). Here, the models have associated the fairness-sensitive attributes (for example, ZIP code and surname) with main prediction task labels (for example,

mortgage loan rejection). At the inference time, the model would amplify the bias and show discrimination toward certain demographic groups, for example, African Americans.

Motivating other directions. We can also take advantage of the insights discussed above to motivate the development of other directions.

Backdoor attack. While we have focused on the setting in which LLMs have unintentionally captured undesirable shortcuts, we must note the adversary can intentionally insert shortcuts into LLMs, which could be a potential security threat to the deployed LLMs. This is called the backdoor attack (or poisoning/Trojan attack). Backdoor attackers insert human-crafted easy patterns that serve as shortcuts during the model training process, explicitly encouraging the model to learn shortcuts. Representative examples include modifying the style of text and adding shortcut unigrams such as double quotation marks.

Watermarking. Unlike malicious use of shortcut learning as the backdoor attack, shortcut learning can also be used for benign purposes. Trigger patterns can be inserted as watermarks by model owners during the training phase to protect the IP of companies. When LLMs are used by unauthorized users, shortcuts in the format of trigger patterns can be used by the stakeholders to claim ownership of the models.

Prompt-Based Paradigm

In previous sections, we have explored the characterization of the shortcut learning problem in the pre-training and fine-tuning training paradigm of medium-sized language models (typically with less than a billion parameters). With the recent emergence of huge-sized language models (with billions of parameters) such as GPT-3 and T5, the prompt-based paradigm has evolved into a new training paradigm with distinct formats from the standard fine-tuning paradigm. Consider the example of prompt for GPT-3. Using natural language instructions and/or demonstration of a few tasks, the LLM can generate the desired output without the need for gradient updates or fine-tuning.

Robustness of prompt-based methods. There are two types of prompt-based paradigms: prompt-based fine-tuning and prompting without fine-tuning. Prompt-based finetuning aims to enable medium-sized language models like BERT or RoBERTa to be few-shot learners, and this still requires optimizing the model's parameters. On the other hand, prompting without fine-tuning is meant for huge-sized language models like GPT-3, where the parameters are fixed and the model is applied to various tasks using different prompts, either discrete or soft.

Prompt-based fine-tuning. Preliminary research has been conducted to examine the shortcut learning challenge in the few-shot prompt-based fine-tuning paradigm.[45] This preliminary study investigated the RoBERTa-large model, which comprises 355 million parameters. This work reveals the following insights: zero-shot prompt-based models exhibit a higher level of robustness against the lexical overlap heuristic during inference, as evidenced by their strong performance on relevant challenge datasets. Conversely, prompt-based fine-tuned models tend to adopt the spurious heuristic as they learn from larger amounts of labeled data, which is reflected by poor performance on OOD datasets. This indicates that prompt-based fine-tuning negatively impacts the robustness and generalizability of a model, just like the standard fine-tuning. The primary reason is that both training methods require adjusting the model's parameters using a biased NLI dataset, leading to a model that heavily relies on dataset biases as shortcuts for predictions.

Prompting without fine-tuning. Preliminary studies are emerging to examine the robustness of prompt-based methods for huge-size language models.[48,52] A study examines the few-shot learning performance of GPT-3 (2.7B, 13B, and 175B parameters) and GPT-2 (1.5B parameters) on text classification and information extraction tasks.[52] The results of the analysis reveal the investigated LLMs are susceptible to majority label bias and position bias, where they tend to predict answers based on the frequency or position of the answers in the training data. Additionally, these LLMs also exhibit common token bias, where they favor answers that are prevalent in their pre-training corpus. Another study explores the impact of prompts on natural language inference tasks in zero-shot and few shot settings using T0 (3B and 11B parameters) and GPT-3 (175B parameters).[48] Experimental results suggest that models can learn just as quickly with many irrelevant or even misleading prompts as they can with effective and instructive prompts. This indicate that models' improvement is not derived from models understanding task instructions in ways analogous to humans' use of task instructions.

Prompting versus standard fine-tuning. GPT-3's few-shot prompt performance is compared to that of BERT and RoBERTa through standard fine-tuning on two natural language inference tasks, that is, MNLI and QQP.[36] Additionally, these models are evaluated on the corresponding difficult OOD datasets: HANS and PAWS. The results show that GPT-3 performs slightly worse in generalization than BERT and RoBERTa on the in distribution MNLI and QQP datasets. On the other hand, GPT-3 achieves higher accuracy on the OOD tests for the majority of testing settings, indicating that GPT-3 has a lower generalization gap between the in-distribution test set and the OOD test set, and thus a higher robustness. However, further analysis of the HANS dataset reveals that GPT-3 still exhibits substantial performance disparities between the bias-supporting and bias-countering subsets. This implies there is room for enhancing the robustness of prompt-based techniques.

Note the current research on prompt-based methods primarily aims at improving LLMs' performance on standard benchmarks. The robustness and generalization of this paradigm still require further investigation. A more thorough evaluation of prompt-based methods is needed and could be a future research topic. Additionally, techniques such as Chain-of-Thought[49] and Scratchpad[24] have been utilized to encourage models to perform intermediate calculations. These methods have proven to enhance the reasoning abilities of LLMs, thus having the potential to improve their robustness and generalization capabilities. Lastly, developing mitigation frameworks that can improve generalization performance on OOD test sets without sac-

rificing standard benchmark performance deserves more attention from the research community.

Conclusion

We present a thorough survey of the LLM's shortcut learning issue for NLU tasks in this article. Our findings suggest that shortcut learning is caused by a skewed dataset, model architecture, and model learning dynamics. We also summarize the mitigation solutions that can be used to reduce shortcut learning and improve the robustness of LLMs. Furthermore, we discuss directions that merit additional research effort from the research community, as well as the connections between shortcut learning and other relevant directions. The key takeaways from this survey's analysis are the current pure data-driven training paradigm for LLMs is insufficient for high-level natural language understanding. In the future, the data-driven paradigm should be combined with domain knowledge at every stage of model design and evaluation to advance the field of LLMs. C

References

1. Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization, 2019; arXiv:1907.02893.
2. Bhargava, P., Drozd, A., and Rogers, A. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the 2nd Workshop on Insights from Negative Results in NLP*, 2021.
3. Branco, R., Branco, A., Silva, J., and Rodrigues, J. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conf. Empirical Methods in Natural Language Processing*.
4. Brown, T.B. et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.
5. Bubeck, S. and Sellke, M. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 2021.
6. Choi, S., Jeong, M., Han, H., and Hwang, S. C2l: Causally contrastive learning for robust text classification. In *Proceedings of the AAAI Conf. Artificial Intelligence 36* (2022), 10526–10534.
7. Clark, C., Yatskar, M., and Zettlemoyer, L. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *Empirical Methods in Natural Language Processing*, 2019.
8. Devlin, J., Chang, M., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Assoc. Computational Linguistics*, 2019.
9. Du, M. et al. Towards interpreting and mitigating shortcut learning behavior of NLU models. *North American Chapter of the Assoc. Computational Linguistics*, 2021.
10. Du, M., Mukherjee, S., Cheng, Y., Shokouhi, M., Hu, X., and Awadallah, A.H. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Annual Meeting of the European Chapter of the Assoc. Computational Linguistics*, 2023.
11. Ethayarajh, K. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Intern. Joint Conf. Natural Language Processing*, 55–65.
12. Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S.R., and Smith, N.A. Annotation artifacts in natural language inference data. *North American Chapter of the Assoc. Computational Linguistics*, 2018.
13. Han, X., Wallace, B.C., and Tsvetkov, Y. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Assoc. Computational Linguistics*, 2020.
14. Jin, D., Jin, Z., Zhou, J.T., and Szolovits, P. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conf. Artificial Intelligence*, 2020.
15. Ko, M., Lee, J., Kim, H., Kim, G., and Kang, J. Look at the first sentence: Position bias in question answering. *Empirical Methods in Natural Language Processing*, 2020.
16. Lai, Y., Zhang, C., Feng, Y., Huang, Q., and Zhao, D. Why machine reading comprehension models learn shortcuts? *ACL Findings*, 2021.
17. Liang, Y., Cao, R., Zheng, J., Ren, J., and Gao, L. Learning to remove: Towards isotropic pre-trained BERT embedding. *Artificial Neural Networks and Machine Learning: Proceedings of the 30th Intern. Conf. Artificial Neural Networks (Bratislava, Slovakia, Sept. 14–17, 2021)*, 448–459.
18. Liu, F. and Avci, B. Incorporating priors with feature attribution on text classification. In *Proceedings of the 57th Annual Meeting of the Assoc. Computational Linguistics*, 2019.
19. Liu, Y. et al. RoBERTa: A robustly optimized BERT pretraining approach, 2019; arXiv:1907.11692.
20. McCoy, R.T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Assoc. Computational Linguistics*, 2019.
21. Mendelson, M. and Belinkov, Y. Debiasing Methods in Natural Language Understanding Make Bias More Accessible. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.
22. Morwani, D., Batra, J., Jain, P., and Netrapalli, P. Simplicity Bias in 1-Hidden Layer Neural Networks, 2023; arXiv:2302.00457.
23. Niven, T. and Kao, H. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Assoc. Computational Linguistics*, 2019.
24. Nye, M. et al. Show your work: Scratchpads for intermediate computation with language models. *Deep Learning for Code Workshop*, 2022.
25. Pezeshkpour, P., Jain, S., Singh, S., and Wallace, B.C. Combining feature and instance attribution to detect artifacts, 2021; arXiv:2107.00323.
26. Pham, T.M., Bui, T., Mai, L., and Nguyen, A. *Out of Order: How important is the sequential order of words in a sentence in natural language understanding tasks?* 2020; arXiv:2012.15180.
27. Prasad, G., Nie, Y., Bansal, M., Jia, R., Kiela, D., and Williams, A. To what extent do human explanations of model behavior align with actual model behavior? In *Proceedings of the 4th Blackbox NLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2021.
28. Qi, F., Chen, Y., Zhang, X., Li, M., Liu, Z., and Sun, M. Mind the style of text! Adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conf. Empirical Methods in Natural Language Processing*.
29. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Machine Learning Research* (2020).
30. Rashid, A., Lioutas, V., and Rezagholizadeh, M. MATE-KD: Masked Adversarial TExt, a companion to knowledge distillation. In *Proceedings of the 59th Annual Meeting of the Assoc. Computational Linguistics and the 11th Intern. Joint Conf. Natural Language Processing*, 2021.
31. Robinson, J., Sun, L., Yu, K., Batmanghelich, K., Jegelka, S., and Sra, S. Can contrastive learning avoid shortcut solutions? *Advances in Neural Information Processing Systems*, 2021.
32. Schuster, T., Shah, D.J., Yeo, Y.J.S., Filizzola, D., Santus, E., and Barzilay, R. Towards debiasing fact verification models. *Empirical Methods in Natural Language Processing*, 2019.
33. Sen, P. and Saffari, A. What do models learn from question answering datasets? In *Proceedings of the 2020 Conf. Empirical Methods in Natural Language Processing*.
34. Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 2020.
35. Shi, Y. et al. Gradient matching for domain generalization. In *Proceedings of the 2022 Intern. Conf. Learning Representations*.
36. Si, C. et al. Prompting gpt-3 to be reliable, 2022; arXiv:2210.09150.
37. Si, C., Wang, S., Kan, M., and Jiang, J. What does BERT learn from multiple-choice reading comprehension datasets? (2019); arXiv:1910.12391.
38. Sinha, K., Jia, R., Hupkes, D., Pineau, J., Williams, A., and Kiela, D. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *Empirical Methods in Natural Language Processing*, 2021.
39. Stacey, J., Belinkov, Y., and Rei, M. Supervising Model Attention with Human Explanations for Robust Natural Language Inference. In *Proceedings of the 2022 AAAI Conf. Artificial Intelligence*.
40. Stacey, J., Minervini, P., Dubossarsky, H., Riedel, S., and Rocktschel, T. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. *Empirical Methods in Natural Language Processing*, 2020.
41. Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the 2017 Intern. Conf. Machine Learning*.
42. Teney, D., Abbasnejad, E., and van den Hengel, A. Unshuffling data for improved generalization, (2020); arXiv:2002.11894.
43. Tu, L., Lalwani, G., Gella, S., and He, H. An empirical study on robustness to spurious correlations using pre-trained language models. *Trans. Assoc. Computational Linguistics* (2020).
44. Utama, P.A., Moosavi, N.S., and Gurevych, I. Towards debiasing NLU models from unknown biases. *Empirical Methods in Natural Language Processing*, 2020.
45. Utama, P.A., Moosavi, N.S., Sanh, V., and Gurevych, I. Avoiding inference heuristics in few-shot prompt-based finetuning. *Empirical Methods in Natural Language Processing*, 2021.
46. Vardi, G., Yehudai, G., and Shamir, O. Gradient methods provably converge to non-robust networks, 2022; arXiv:2202.04347.
47. Wang, B. et al. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings for the 35th Conf. Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
48. Webson, A. and Pavlick, E. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conf. North American Chapter of the Assoc. Computational Linguistics: Human Language Technologies*.
49. Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.
50. Yang, J., Gupta, A., Upadhyay, S., He, L., Goel, R., and Paul, S. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Assoc. Computational Linguistics*, 2022.
51. Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conf. Empirical Methods in Natural Language Processing*.
52. Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 2021 Intern. Conf. Machine Learning*, 12697–12706.

Mengnan Du is an assistant professor in the Department of Data Science at New Jersey Institute of Technology, University Heights, Newark, NJ, USA.

Fengxiang He is a lecturer in the school of Informatics at the University of Edinburgh, Scotland.

Na Zou is an assistant professor of engineering technology and industrial distribution at Texas A&M University, College Station, TX, USA.

Dacheng Tao is a professor of computer science at the University of Sydney, Australia.

Xia Hu is an associate professor of computer science at Rice University, Houston, TX, USA.