Supervised and Unsupervised Probing of Shortcut Learning: Case Study on the Emergence and Evolution of Syntactic Heuristics in BERT

Elke Vandermeerschen
Faculty of Engineering Science
KU Leuven
elke.vandermeerschen@kuleuven.be

Miryam de Lhoneux
Department of Computer Science
KU Leuven
miryam.delhoneux@kuleuven.be

Abstract

Contemporary language models (LMs) such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2023), GPT-4 (OpenAI, 2023), have exhibited remarkable capabilities, effectively addressing long-standing challenges in the field. However, these models rely on shortcut learning, using a decision rule that relies on superficial cues that are spuriously correlated with the labels (Geirhos et al., 2020). In this research, we focus on the reliance on a specific type of shortcuts, namely syntactic heuristics, in BERT when performing Natural Language Inference (NLI), a representative task in Natural Language Understanding (Jeretic et al., 2020). By making use of two probing methods, one supervised, one unsupervised, we investigate where these shortcuts emerge, how they evolve and how they impact the latent knowledge of the LM. Our findings reveal that syntactic heuristics are absent in pretrained models but emerge and evolve as the model is finetuned with datasets of increasing size. The adoption of these shortcuts varies across different hidden layers, with specific layers closer to the output contributing more to this phenomenon. Despite the model's reliance on shortcuts during inference, it retains information relevant to the task, and our supervised and unsupervised probes process this information differently.

1 Introduction

Probing the dynamics of shortcut learning. Contemporary neural network-based language models (LMs) such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2023), GPT-4 (OpenAI, 2023), have exhibited remarkable capabilities, effectively addressing long-standing challenges in the field. Despite these achievements, and the development of increasingly larger models that excel on various benchmarks, certain limitations and risks have recently demanded increased attention. High scores on standard benchmarks do not necessarily indicate that a model actually possesses the underlying generalization capabilities and knowledge essential for such performance (Geirhos et al., 2020; Du et al., 2023; Browning and LeCun, 2023; McCoy et al., 2019). For instance, Browning and LeCun demonstrate that the outstanding performance of LMs on the Winograd Schema Challenge is not driven by a genuine understanding of language, world knowledge and common sense reasoning, capabilities originally deemed necessary to solve the challenge. Across various fields and applications, researchers have observed that while a model may achieve impressive results on standard benchmarks, this performance does not generalize well across different datasets. Specifically, there is a substantial drop in performance on out-of-distribution (o.o.d.) datasets.

Geirhos et al. identify this as a manifestation of shortcut learning, where models rely on superficial strategies that work well on standard benchmarks but fail when faced with more complex testing scenarios. Essentially, shortcut learning occurs when models rely on surface-level cues to complete tasks, instead of learning to reason about the task.

The problem of shortcut learning in LMs can result from many different factors, including the limitations of the training dataset, the model architecture and size, pretraining objectives and finetuning procedures (Du et al., 2023). In this paper, we investigate shortcuts used in Natural Language Inference (NLI) tasks, where the aim is to determine if one sentence (hypothesis) is entailed by another sentence (premise). By combining two probing methods, one supervised, one unsupervised, we investigate where these shortcuts emerge, how they evolve and how they impact the latent knowledge of the LM.

Research questions and contributions. To investigate the adoption and development of these syntactic heuristics and examine their influence on the knowledge embedded within the hidden repre-

Findings of the Association for Computational Linguistics: ACL 2025, pages 9592–9604
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

sentations, we address the following research questions:

RQ1 At what point does shortcut learning begin to manifest? Can we identify traces of it in the representations of the pretrained model, or only after a certain amount of finetuning?

RQ2 How does shortcut learning evolve with increased finetuning dataset size, and across the different hidden layers?

RQ3 How does shortcut learning modify the latent knowledge of the model? Is there a discrepancy between the model's outputs and the underlying knowledge encoded in the model?

To address these questions, we combine a supervised and unsupervised probe.

The contributions of this study are threefold. First, it presents an empirical case study on the emergence and evolution of syntactic heuristics in BERT. Second, it introduces a methodological innovation for model analysis: a novel combination of supervised and unsupervised probes, enabling us to evaluate how the learning capacity and bias of the supervised method influence the scores. Finally, it provides evidence supporting the validity of an experimental method, Contrast Consistent Search (CCS, Burns et al., 2024) as an unsupervised probe, as outlined in the methodology section 3.

2 Related work

Syntactic heuristics Solving NLI tasks typically requires syntactic and semantic understanding, common sense reasoning and world knowledge. However, Niven and Kao (2019), McCoy et al. (2019), Geirhos et al. (2020), Du et al. (2023) and Hartmann et al. (2021) demonstrated that LMs finetuned on NLI datasets fail to build on these requirements, and instead rely on the use of shortcuts. Consequently, they have learned to solve the dataset instead of solving the task (Du et al., 2023). Among these shortcuts are the syntactic heuristics identified by McCoy et al.: the lexical overlap heuristic, subsequence heuristic and constituent heuristic. The lexical overlap heuristic assumes that a premise entails all hypotheses constructed from words in the premise.[1] The subsequence heuristic suggests that a premise entails all its contiguous subsequences.[2] Finally, the constituent heuristic asserts that a premise entails all complete subtrees within its parse tree.[3]

These heuristics achieve (misleadingly) high accuracies on the MNLI test set but fail in many o.o.d. cases. To assess this, McCoy et al. developed a controlled evaluation set called HANS (Heuristic Analysis for NLI Systems), where these heuristics fail. Their findings indicate that models finetuned on MNLI perform poorly on the HANS challenge set, suggesting they rely on the proposed heuristics. Notably, BERT's poor accuracy on the HANS set is remarkable given that BERT has access to relevant syntactic information (Hewitt and Manning, 2019).

Probing the dynamics of shortcut learning Our approach focuses on the layerwise evolution of shortcuts and uses a combined methodology of supervised and unsupervised probing. It builds upon the foundational studies on the reliance on shortcuts in NLI, as demonstrated in McCoy et al. (2019), as well as broader research on shortcut learning in LMs (Geirhos et al., 2020; Branco et al., 2021; Ray Choudhury et al., 2022; Du et al., 2023). More recently, Zhang et al. (2024) have probed causality manipulation hierarchically by introducing different shortcuts to models and observing their behaviors, while Tang et al. (2023) have investigated LMs' reliance on shortcuts or spurious correlations within prompts. Sun et al. (2024) demonstrate how more recent, larger generative models exploit spurious correlations for predictions.

Unsupervised probing for latent knowledge For our unsupervised probe, we build on the work of Burns et al. (2024) and use Contrast Consistent Search (CCS). CCS is built around a key characteristic of the notion of truth: logical consistency, a sentence and its negation cannot both be true. Assuming that a model internally evaluates the truth value of an input, this can in principle be recovered from the hidden representations. CCS learns a linear transformation of the hidden representations such that the transformed space embeds this logical consistency (Burns et al., 2024). The input sentences are converted into contrast pairs, a sentence

[1] For example, the lexical overlap heuristic would correctly assume that the premise 'The pupil was asked by the teacher' entails the hypothesis 'The teacher asked'. However, this hypothesis would incorrectly assume that the premise entails the hypothesis 'The pupil asked the teacher'.

[2] For instance, this heuristic would correctly assume that 'The experienced nurses working with the doctors refused' entails 'Nurses working with the doctor refused', but incorrectly assume it equally entails 'The doctors refused'.

[3] The constituent heuristic would, correctly assume that the premise 'If the accountants knew, they wouldn't have resigned' entails 'They wouldn't have resigned', but equally that this entails that 'The accountants knew'.

and its negation,[4] and passed through the model. The learned linear projection is applied to the generated hidden representations, enabling the model to predict opposite labels for the two sentences of the pair. This classification reflects whether the representations correspond to a true or false statement. CCS scores quantify the number of statements that are correctly identified as 'true' or 'false'.

3 Methodology

This case study focuses on a single model, a specific finetuning dataset, and one challenge set, enabling a controlled analysis of both the method's effectiveness and a concrete instance of shortcut learning. We conduct the study on BERT, an open-source LM widely used and studied, particularly in interpretability research, where the field of "BERTology" examines its behavior and intermediate representations in relation to linguistic properties (Rogers et al., 2020). Given its prominence, BERT is a natural starting point for testing this new methodological paradigm.

Concretely, in a first step we finetune BERT (Devlin et al., 2019) with a binarized version ('entailment' and 'contradiction' examples only) of MultiNLI (Williams et al., 2018). Following finetuning, we convert MNLI and HANS examples into contrast pairs.[5] These contrast pairs are fed into the finetuned models, and the accuracy scores of our supervised probe, Logistic Regression (LR), and usupervised probe, CCS, are measured.

To obtain LR scores, we use the representations of the contrast pairs, along with the true labels. The hidden states are split into train and test and LR accuracy is measured on the test set. Having access to the labels makes this approach similar to a regular supervised probe.

By comparing the -supervised- LR accuracies for MNLI and HANS examples, we can determine whether the model relies on the syntactic heuristics. Since these heuristics fail on HANS examples, consistently lower scores on the HANS examples than on MNLI examples provide evidence for the reliance on those heuristics. The gap between both scores (LR MNLI versus LR HANS) can be taken as a measure of the reliance on syntactic heuristics. We examine this gap in models fine-tuned with datasets of different sizes and measure the progression of this difference across the hidden layers. We identify which layers have the most significant impact on the adoption of syntactic heuristics and whether the presence of these heuristics increases towards the layers closer to the output layer.

To obtain CCS scores, we use the same contrast pair representations without the labels, and apply the learned linear projection, enabling the model to predict opposite labels for the two sentences of the pair. CCS scores quantify the number of statements that are correctly identified as 'true' or 'false'. As such, the gap between the HANS and MNLI based CCS scores indicate whether the syntactic heuristics are inherently encoded within the hidden representations. Thereby we demonstrate that CCS can resolve NLI questions in an unsupervised manner, supporting its effectiveness as a probing method. Additionally, we demonstrate the complementary nature of CCS and LR. With a supervised probe like LR, the performance could be attributed to the probe's learning capacity. By contrast, CCS is unsupervised, which allows us to access information inherently present in the model's representations. CCS shows which patterns are genuinely latent versus those that only appear through probe training. The final research question explores the extent to which the learned heuristics have altered the model's internal knowledge and its representation of the *truth*[6] of the input sentences. To address this question, we examine the knowledge in the hidden representations of the finetuned models, reflected in the gap between MNLI and HANS LR and CCS scores, and compare this to inference results. CCS was initially conceived as a prototype to explore unsupervised methods for identifying the *beliefs*[7] of a model (Burns et al., 2024). Comparing CCS to LR and inference results, enables us to investigate whether a significant gap exists between the knowledge the model possesses and what it outputs.

[4] A contrast pair consists of the same question presented twice, once with the answer 'yes' and once with the answer 'no'. To translate a premise and hypothesis into these contradicting statements we make use of PromptSource Templates (Bach et al., 2022).

[5] This is necessary to obtain CCS scores, as explained in section 2.

[6] In the context of this paper and evaluating CCS on NLI, it is reasonable to accept that the *truth* coincides with the labels, an assumption that is also used in the evaluation of the CCS accuracy scores. In other contexts this might not be as straightforward. We use the italic font to indicate this ambiguity.

[7] The term *beliefs* here and further in this paper, is by no means intended to refer to mental states in an anthropomorphic way, rather as the model 'has a representation of the validity and truthfulness of a statement, reflected in the hidden representations'. We follow the terminology used by (Burns et al., 2024).

4 Experiment and Results

To examine whether the degree of reliance on syntactic heuristics changes with the size of the dataset used for finetuning, we finetuned BERT[8](Devlin et al., 2019) progressively.

4.1 Experimental setup

Finetuning datasets are binarized versions of MNLI containing 25, 100, 1K, 8K and 16K examples. Additionally, we finetuned the model on a mixed dataset, containing 8K examples from MNLI and 8K from HANS.[9] The following finetuning parameters were used: learning rate 2e-05, train batch size 8, Optimizer Adam with betas 0.9,0.999 and epsilon 1e-08 and number of epochs 5.[10]

4.2 Preliminary experiments

Preliminary experiments were strictly used to identify and select probing parameters to ensure the probing methods are both robust and informative. The appropriate values for the following parameters are determined: first, the prompts used to create contrast pairs, specifically the prompt 'Does it follow that', from the PromptSource HANS template (Bach et al., 2022). To compare LR and CCS accuracies across different finetuned models, we consistently use the hidden representation from layer 23—identified as the most informative for LR in our initial experiments—to ensure comparability across methods, even though CCS scores peak slightly earlier. This approach aligns with prior work showing that BERT's upper layers encode increasingly semantic and task-specific information (Tenney et al., 2019; Rogers et al., 2020). Finally, it is determined that 1.000 examples are needed to sufficiently train the CCS probe.

To decide which sentence representation to use, we compared results obtained with mean pooling, last token representation, and the [CLS] token (Appendix Table1). While mean pooling results in the highest LR values, the CCS performance is remarkably low, barely exceeding chance level. Therefore, we use the [CLS] token, which results in more informative CCS scores, while still obtaining relatively high LR scores.

4.3 Results based on progressive finetuning

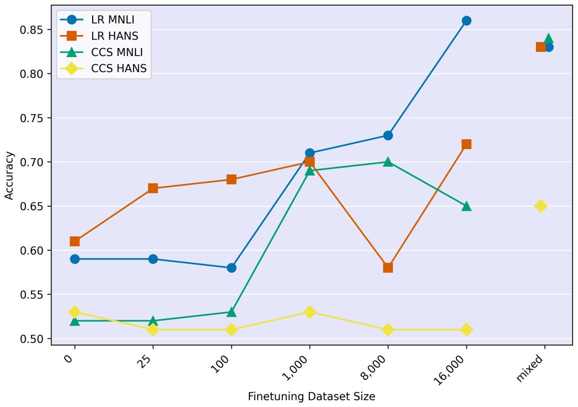

Figure 1: LR and CCS scores as a result of finetuning

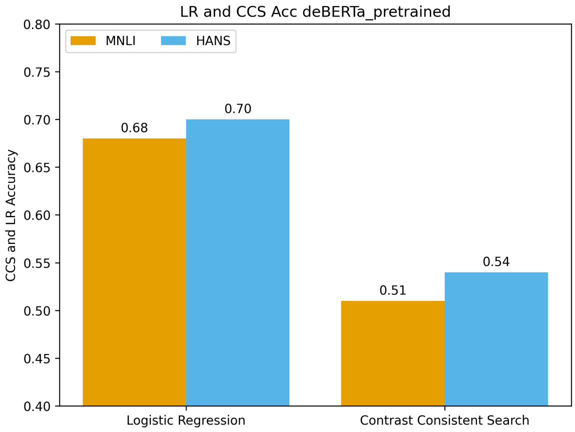

Figure 2: LR and CCS deBERTa pretrained-only

4.3.1 Results from pretrained models

The average **LR** accuracy from the hidden representations (layer 23) of BERT (pretrained-only) is 0.59 for MNLI contrast pairs. In comparison, the accuracy is slightly higher for HANS examples, at 0.61 (Figure 1, Finetuning Dataset Size 0). The **CCS** accuracy derived from the same hidden representations are relatively low and quite similar for MNLI and HANS examples, hovering just above 0.5.

The higher scores on HANS examples -where the heuristics fail- suggest that the pretrained model does not rely on syntactic heuristics as a decision rule for NLI. However, since the scores are low and close to chance level, the representations might simply not be informative enough, thus this preliminary conclusion needs to be verified.

[8] Bert-large-cased, https://huggingface.co/google-bert/bert-large-cased, configuration: 24 layers, 1024 hidden dimensions, 16 attention heads 336M parameters

[9] Examples from the HANS dataset are selected from all categories, challenging the different heuristics and subcases.

[10] We opted for a fixed number of epochs to ensure consistent exposure across dataset sizes, ensuring the model has an equal opportunity to see each example in the dataset a fixed number of times, regardless of dataset size. This is particularly important as we start with (very) small datasets.

Therefore, we examined the hidden representations from deBERTa (He et al., 2021) pretrained-only (Figure 2). The LR accuracy scores derived from the hidden representations of deBERTa are substantially higher than those from BERT, 0.68 on MNLI examples and 0.70 on HANS examples. The LR and the CCS scores on MNLI are comparable to those on HANS, with a slight advantage for HANS examples, mirroring the results observed with BERT pretrained. This consistency reinforces the conclusion that there is no evidence of syntactic heuristics being adopted during pretraining.

4.3.2 Results from models finetuned with MNLI

Figure 1 illustrates that finetuning BERT on a small number of MNLI datapoints does not enhance the **LR** scores on MNLI. However, LR scores on HANS increase, reaching values of 0.67 (25 examples), 0.68 (100 examples), and thus LR scores on HANS-based examples remain slightly higher than those on MNLI examples.[11] As of finetuning with over 1K examples, MNLI LR scores start increasing, in contrast to the sharp decrease in HANS LR scores observed between the models finetuned with 1K and 8K examples.[12] Since the syntactic heuristics fail on the HANS examples, the difference in LR accuracies between MNLI and HANS-based examples serves as evidence of the model's reliance on shortcuts. Therefore the observed gap between the two scores, reflects the emergence of shortcuts. Finetuning with even more examples (16K) leads to a sharper increase of MNLI-based LR scores, reaching a maximum of 0.86 compared to 0.72 for HANS-based LR scores.

The **CCS** scores measured on MNLI examples only exceed chance level as of finetuning with at least 1K examples, reaching values of 0.69 (1K finetuning examples), 0.7 (8K examples) and 0.65 (16K examples). In contrast, the CCS scores on the basis of HANS examples remain close to chance level, even for the models finetuned with several thousand examples.

4.3.3 Results from the model finetuned with the mixed dataset

Figure 1 additionally illustrates the results from the model finetuned with a balanced mix of MNLI and HANS examples. **LR** scores for HANS and MNLI examples are quite similar (0.85 vs 0.83), indicating that this model does not rely on the syntactic heuristics as the primary decision rule.

Examining the **CCS** scores of this model, we observe that the accuracy for HANS-based CCS, measured at layer 23, is still considerably lower than for MNLI-based CCS (0.62 vs. 0.84). Despite this gap, there is a notable increase in HANS-based CCS scores compared to those of the other models finetuned with MNLI examples only, where HANS-based CCS scores remain close to chance level.

These results clearly indicate the emergence and development of shortcuts as a result of finetuning, as we will discuss further in section 5.

4.4 Results across the hidden layers

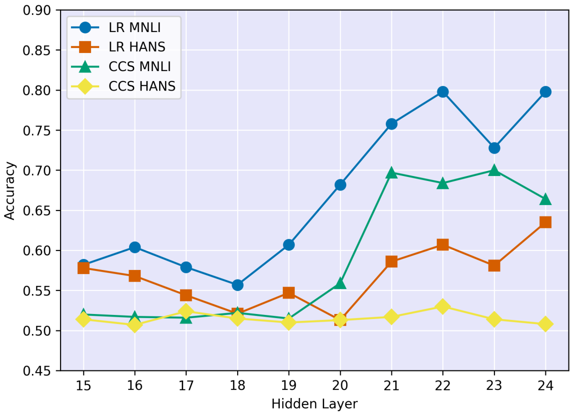

Figure 3: LR and CCS evolution across the hidden layers BERT finetuned with 8K examples

4.4.1 Results across the hidden layers from models finetuned with MNLI

Figures 3 and 4 illustrate the evolution of LR and CCS accuracy scores measured across the hidden layers of the two models finetuned on the largest datasets (8K and 16K finetuning examples). The results reveal certain similarities: accuracy values remain relatively low (under 0.6) throughout the first 19 layers.[13]

From layer 20 onward, we see a strong increase in the MNLI-based LR and CCS scores,

[11] Results mentioned are averages of 5 runs with 2K examples per run. For detailed results, please refer to Appendix A.2

[12] This sharp decrease is unexpected and remains difficult to fully explain. Repeated experiments consistently show the same pattern. However, it is important to note that these results are measured at layer 23. The decrease is less pronounced when examining other layers, such as layer 24.

[13] Note that Figure 3 and Figure 4 show only the last 10 layers; for results spanning all hidden layers, refer to Appendix A.3.

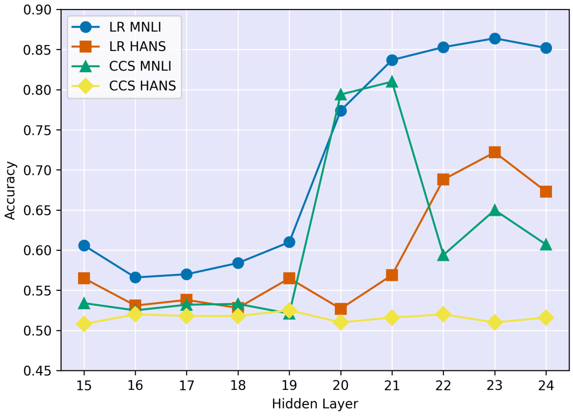

Figure 4: LR and CCS evolution across the hidden layers BERT finetuned with 16K examples

whereas HANS-based LR scores show only a limited increase. The difference between MNLI- and HANS-based LR scores, reflecting the syntactic heuristics, is most pronounced in the final 5 layers. The HANS-based CCS values remain close to 0.5 across all layers of both models.

Despite these important similarities, the results reveal striking differences in how LR and CCS scores evolve across the hidden layers of the two models. Firstly, while the CCS scores of the model finetuned with 8K examples remain relatively stable during the last 4 layers, those of the 16 K model demonstrate a sharp drop at layer 22.[14] Secondly, most of the maximum scores are reached at prefinal layers, but the specific layers differ: for the 8K model, they occur at layer 24 (LR) and layers 21–23 (CCS), whereas for the 16K model, they are found at layer 23 (LR) and layer 21 (CCS). These differences will be examined in more detail in 5.2.

4.4.2 Results across the hidden layers from the model finetuned with the mixed set

Focusing on the evolution across the hidden layers of the model finetuned with the mixed dataset (Figure 5), we see a different picture: as expected, there is no gap between the MNLI and HANS **LR** scores. The HANS-based **CCS** scores for this model finetuned with the mixed dataset are substantially higher than those of the MNLI-only finetuned models, but remain well below its MNLI-based CCS scores.

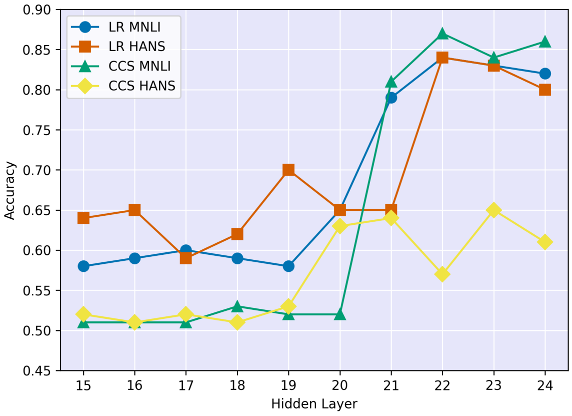

Figure 5: LR and CCS evolution across the hidden layers BERT finetuned with the mixed dataset

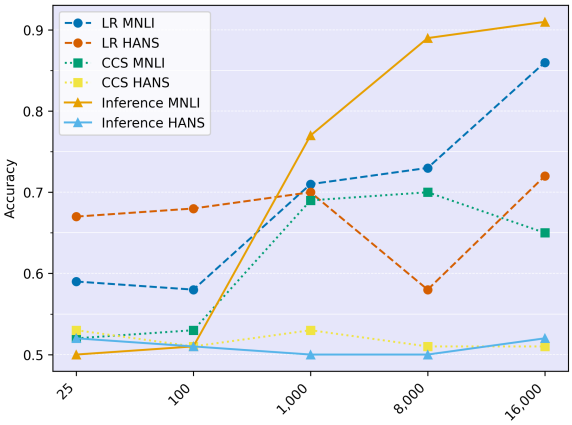

Figure 6: Inference versus LR and CCS scores

4.5 Inference results vs. LR and CCS

Figure 6 illustrates the evolution of inference scores, alongside LR and CCS, on MNLI and HANS examples, based on progressive finetuning.[15] Focusing on the difference between MNLI and HANS-based scores, we observe a notable increase of this difference, particularly in the inference scores. Finetuning with more MNLI data consistently enhances (inference) performance on MNLI examples, while only marginally improving results on HANS examples.

In contrast with the inference scores, the clear increase of HANS LR scores of our model finetuned with the biggest dataset is remarkable, reaching a maximum score of 0.72 (16K finetuning examples). However, CCS scores for HANS examples -similar to the inference scores- barely exceed chance level.

[14] The sudden drop in CCS results for the model finetuned with 16K examples is surprising. It may indicate that deeper layers adapt to MNLI-specific parameters based on heuristic cues, complicating true/false classification. However, similar patterns are less pronounced in other models, where CCS increases stop around layer 22 but without a comparable drop.

[15] We consistently use the LR and CCS accuracy scores measured on the hidden representations of layer 23, where the scores are usually relatively high, but not necessarily the maximal scores for both LR and CCS for all models.

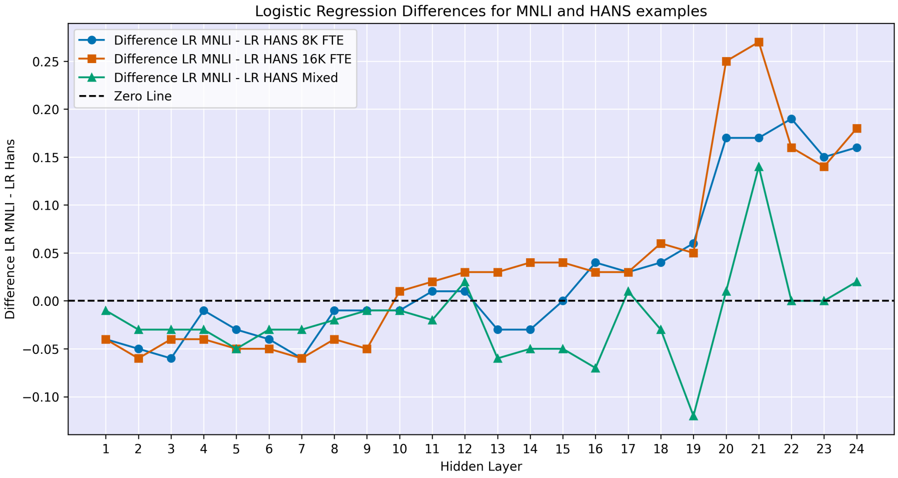

Figure 7: Differences between LR accuracies for MNLI and HANS Examples

5 Analysis and Discussion

5.1 Adoption of syntactic heuristics as a result of finetuning

The results presented in 4.3.1, with higher accuracy scores for HANS than for MNLI examples, demonstrate that the investigated syntactic heuristics are absent in the pretrained models. Although NLI classification is a challenging task for BERT, its hidden representations contain some relevant information. In a supervised manner, this information can be leveraged, to a certain extent, to classify MNLI and HANS examples, resulting in LR scores for both datasets around 0.6. As the pretrained model is trained on extensive and diverse datasets to develop a rich contextualized understanding and representation of language, it is unsurprising that we do not observe the reliance on these specific syntactic heuristics within this model. It is not incentivized to learn decision rules tailored to any particular dataset.

Analyzing the results obtained with the finetuned models (4.3.2, Figure 1), we observed a stronger increase in MNLI-based **LR** scores compared to the limited rise in HANS-based LR scores. This results in a widening gap between the scores on both datasets, thereby demonstrating the model's increasing reliance on syntactic heuristics, providing strong evidence that finetuning with larger datasets leads to a greater reliance on the syntactic heuristics. The shortcuts begin to emerge when the finetuning dataset contains more than 1K examples. This trend is also reflected in the results from **CCS**, our unsupervised probe: for MNLI examples, finetuning leads to more informative representation, enabling the construction of a linear transformation that reflects the truth value of a statement. This is not the case for HANS-based contrast pairs: their CCS scores do not benefit from finetuning on more MNLI examples. Consequently, the gap between CCS scores on MNLI and HANS widens. Furthermore, the results obtained with the mixed dataset model (4.3.3) provide additional support for CCS as an effective unsupervised probing method: the higher CCS scores on HANS examples achieved by the mixed dataset model demonstrate that obtaining above-chance performance on HANS is indeed possible. This contrasts with the low HANS CCS scores seen in models finetuned solely on MNLI. This supports the conclusion that the low scores observed in those models are caused by their reliance on syntactic heuristics.

The increased adoption of the heuristics by the MNLI-only finetuned models coincides with the improved performance on MNLI examples. This suggests that the model's enhanced performance is largely due to a developed decision rule based on these shortcuts, learned from the dataset. Consequently, finetuning with diverse datasets -including challenging examples- may help to prevent the adoption of shallow heuristics. Our combined results with the model finetuned with the mixed dataset (Figure 1) illustrate how incorporating challenging examples can mitigate the adoption of shallow heuristics. For HANS examples, we observed a substantial increase in all accuracy measures (LR, CCS, and Inference) compared to models finetuned exclusively on MNLI.

5.2 Evolution of the shortcuts across the hidden layers

The differences in the evolution of **LR** values in the two models investigated (4.4.1, Figure 3 and Figure 4) suggest that, although similar information may be encoded, it is distributed across different layers in each model. These differences indicate varying reliance on shortcuts within the two models, as reflected in the scores obtained with our supervised probe at identical layers. Focusing on the reliance on shortcuts as illustrated by the gap between MNLI and HANS based LR results (Figure 7), we even observe negative values in the first 10 layers. This indicates that, prior to the transformations across layers and the construction of the learned decision rule, HANS examples are easier to solve compared to MNLI examples. Certain layers contribute more to the development of the heuristics, most salient in a steep increase on MNLI examples, compared to a limited increase on HANS examples. The layers contributing most to these heuristics are those close to the output layer, though typically not the final hidden layer. This suggests that the reliance on shortcuts is not exclusively output-driven.[16] Similarly, the analysis of **CCS** scores across layers in the two models (Figures 3 and 4) identified important differences, indicating that the increased adoption of the syntactic heuristics in the representation of the truth of a sentence occurs in different layers for each model. Analysis of the layerwise LR scores from the model finetuned with the mixed dataset (4.4.2) shows no reflection of the heuristics in the 3 final layers, despite a limited gap at layer 21 (Figure 7). However the unsupervised CCS scores on MNLI are consistently higher than on HANS examples. Examining the evolution across the layers revealed that our supervised and unsupervised probes capture different knowledge in the hidden representations, highlighting their complementarity. The differences between the LR and CCS results obtained on specific layers illustrate that the two probes capture NLI-relevant information in different ways, and build on different aspects of what is encoded in the representations. Some signals may be more useful for LR, where the probe can leverage its learning capacity to extract predictive patterns — even if these are less explicitly structured. CCS, on the other hand, relies on internal logical consistency in the representation space, which may be stronger in slightly earlier layers (before task-specific patterns dominate).

5.3 Impact on the latent knowledge

While our finetuned models fully rely on syntactic heuristics as the main decision rule during **inference**, resulting in HANS-based inference scores at chance level, this is not reflected in the LR values (4.5). The increase in LR scores on HANS examples with more finetuning suggests that finetuning on MNLI provides the model with some relevant knowledge to classify HANS examples. This is not reflected in the inference scores, where the learned decision rule completely fails on the HANS examples.

Finetuning on MNLI examples suppresses the capability of performing inference on HANS examples, while some relevant information in the hidden representations of certain layers is still present. This information proves to be relevant for classifying contrast pair examples, but only in a supervised way. The relatively high LR scores on HANS examples, despite very low inference scores, suggest the model retains relevant information in the certain hidden layers. However, it fails to produce the correct answer for examples where the heuristics break down. This suggests the model's internal knowledge is not entirely overridden by the syntactic heuristics.

The striking contrast between the high LR scores and low CCS and inference scores on HANS pairs suggests that while the supervised probe builds on relevant knowledge less affected by the heuristics, this may not hold for the unsupervised approach. The low unsupervised CCS and inference scores indicate that shortcut reliance may be even stronger than supervised LR scores imply. This highlights the supervised probe's role in reconstructing underlying knowledge, which may be represented in hidden layers but not always observable in outputs. Consequently, the learning capacity of the supervised probe could be a critical factor in interpreting the observed outcomes (Pimentel et al., 2020; Singh et al., 2024).

[16] If the shortcut reliance were exclusively output-driven, in a sense of mimicking the training data, we would expect a gradual increase towards the final hidden layer. However, here we observe their emergence and accumulation across various layers of the model.

6 Conclusions

In this research, we traced the evolution of the adoption of syntactic heuristics as a decision rule for NLI classification, adopted by BERT, finetuned on MNLI datasets. To identify the learned heuristics, we used a challenge dataset, HANS, and combined supervised and unsupervised probing methods. Our findings demonstrate the progressive adoption of these shortcuts as a result of finetuning dataset size, observable across the model's hidden layers, with specific layers closer to the output contributing more to this phenomenon. Our study illustrates the need for finetuning with diverse datasets, including examples where shallow heuristics fail. A deeper understanding of the inductive biases of LMs and ongoing research into o.o.d. challenge datasets can aid in ensuring the diversity of training data.

Despite the model's reliance on shortcuts during inference, it retains information relevant to the task, and our supervised and unsupervised probes process this information differently. A key challenge remains in effectively leveraging this latent knowledge and encouraging the model to reason over it. Further research is needed to uncover the underlying mechanisms and explore strategies to harness this knowledge. One of the key contributions of this paper is therefore methodological: integrating results obtained with supervised and unsupervised probing with inference results mitigates individual limitations of these approaches.

Limitations

In this research, we focused on BERT (Devlin et al., 2019), a relatively small and widely studied model. This choice allows for controlled experimentation and comparability with existing probing research, but it also strongly limits the scope of our findings. The observed shortcut behaviors and the patterns revealed through the combination of supervised and unsupervised probing may not directly generalize to larger or more recent architectures. However, even larger generative models might rely on shallow heuristics in a similar manner (Du et al., 2023). Extending our approach to more recent and diverse models would help assess the robustness of our findings across architectures with different inductive biases, training objectives, and internal representation structures. This would require adapting the probing setup to account for variations in architecture (e.g., encoder-only vs. encoder-decoder), layer depth, and pretraining strategies.

We examined the effects of full finetuning on the development of shortcuts in LMs and did not explore less extensive task adaptation techniques. Parameter-efficient finetuning methods, such as the use of adapter modules (Houlsby et al., 2019), and their impact on shortcut learning were beyond the scope of this research, but would be an interesting extension of this work.

Another limitation of this study is that we assessed reliance on shortcuts by considering lexical overlap, subsequence, and constituent heuristics as a whole, rather than isolating and investigating the specific impact of each heuristic.

Additionally, future work can look at more datasets and heuristics to confirm the findings of this case study.

Acknowledgments

This paper is based on the Master's Thesis submitted to obtain the degree of Advanced Master of Science in Artificial Intelligence, Faculty of Engineering Science, KU Leuven (Vandermeerschen, 2024). The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government. We are deeply grateful to Torben Swoboda for his philosophical reflections on the question of alignment, which informed the background of this work and helped us reflect on its broader challenges and implications. We would also like to thank Tim Van de Cruys and Bram Vanroy for their critical reading and thought-provoking questions, which helped strengthen our research.

References

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Browning and Yann LeCun. 2023. Language, common sense, and the winograd schema challenge. *Artificial Intelligence*, 325:104031.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2024. Discovering latent knowledge in language models without supervision. *Preprint*, arXiv:2212.03827.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Preprint*, arXiv:2208.11857.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Mareike Hartmann, Miryam de Lhoneux, Daniel Hershcovich, Yova Kementchedjhieva, Lukas Nielsen, Chen Qiu, and Anders Søgaard. 2021. A multilingual benchmark for probing negation-awareness with minimal pairs. In *Conference on Computational Natural Language Learning*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, 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.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Sagnik Ray Choudhury, Anna Rogers, and Isabelle Augenstein. 2022. Machine reading, fast and slow: When do models "understand" language? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 78–93, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *Preprint*, arXiv:2402.01761.

Zechen Sun, Yisheng Xiao, Juntao Li, Yixin Ji, Wenliang Chen, and Min Zhang. 2024. Exploring and mitigating shortcut learning for generative large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6883–6893, Torino, Italia. ELRA and ICCL.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4645–4657, Toronto, Canada. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Elke Vandermeerschen. 2024. Tracing the evolution of syntactic heuristics and their impact on the latent knowledge of large language models. Master's thesis, KU Leuven.

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.

Chenyang Zhang, Haibo Tong, Bin Zhang, and Dongyu Zhang. 2024. Probing causality manipulation of large language models. *Preprint*, arXiv:2408.14380.

A Appendix

A.1 Results preliminary experiments to determine sentence representation

Sentence Representation	FTE	LR	CCS
Mean pooling	8,000	0,73	0,50
Mean pooling	16,000	0,86	0,50
Last token	8,000	0,84	0,62
Last token	16,000	0,81	0,52
CLS token	8,000	0,73	0,70
CLS token	16,000	0,86	0,65

Table 1: Preliminary Results with different sentence representations, dataset MNLI.

A.2 Detailed results of several runs

Examples	Dataset	LR 1	LR 2	LR 3	LR 4	LR 5	LR Average
25	MNLI	0,598	0,540	0,586	0,624	0,622	**0,59**
100	MNLI	0,574	0,544	0,586	0,576	0,632	**0,58**
1,000	MNLI	0,694	0,708	0,714	0,706	0,734	**0,71**
8,000	MNLI	0,730	0,736	0,724	0,738	0,712	**0,73**
16,000	MNLI	0,852	0,858	0,876	0,858	0,874	**0,86**
16,000	MIXED	0,838	0,874	0,828	0,824	0,798	**0,83**

Table 2: MNLI LR Results based on progressive finetuning 5 runs with 2K examples per run.

Examples	Dataset	LR 1	LR 2	LR 3	LR 4	LR 5	LR Average
25	HANS	0,684	0,66	0,654	0,672	0,670	**0,67**
100	HANS	0,724	0,668	0,670	0,644	0,688	**0,68**
1,000	HANS	0,694	0,702	0,696	0,688	0,732	**0,70**
8,000	HANS	0,566	0,595	0,596	0,582	0,566	**0,58**
16,000	HANS	0,738	0,658	0,748	0,748	0,718	**0,72**
16,000	MIXED	0,822	0,844	0,826	0,840	0,834	**0,83**

Table 3: HANS LR Results based on progressive finetuning 5 runs with 2K examples per run.

Examples	Dataset	CCS 1	CCS 2	CCS 3	CCS 4	CCS 5	CCS Average
25	MNLI	0,532	0,528	0,510	0,513	0,504	**0,52**
100	MNLI	0,52	0,546	0,522	0,526	0,55	**0,53**
1,000	MNLI	0,658	0,690	0,682	0,712	0,722	**0,69**
8,000	MNLI	0,704	0,724	0,576	0,760	0,734	**0,70**
16,000	MNLI	0,762	0,544	0,876	0,522	0,546	**0,65**
16,000	MIXED	0,812	0,876	0,846	0,868	0,810	**0,84**

Table 4: MNLI CCS Results based on progressive finetuning 5 runs with 2K examples per run.

Examples	Dataset	CCS 1	CCS 2	CCS 3	CCS 4	CCS 5	CCS Average
25	HANS	0,538	0,51	0,520	0,538	0,538	**0,53**
100	HANS	0,512	0,506	0,508	0,508	0,52	**0,51**
1,000	HANS	0,518	0,504	0,532	0,55	0,53	**0,53**
8,000	HANS	0,516	0,505	0,514	0,504	0,532	**0,51**
16,000	HANS	0,511	0,508	0,506	0,520	0,506	**0,51**
16,000	MIXED	0,646	0,602	0,672	0,672	0,654	**0,65**

Table 5: HANS CCS Results based on progressive finetuning 5 runs with 2K examples per run.

A.3 Results across all hidden layers

Layer	LR MNLI	LR HANS	CCS MNLI	CCS HANS
24	0,80	0,64	0,66	0,51
23	0,73	0,58	0,70	0,51
22	0,80	0,61	0,68	0,53
21	0,76	0,59	0,70	0,52
20	0,68	0,51	0,56	0,51
19	0,61	0,55	0,52	0,51
18	0,56	0,52	0,52	0,52
17	0,58	0,54	0,52	0,52
16	0,60	0,57	0,52	0,51
15	0,58	0,58	0,52	0,51
14	0,52	0,56	0,51	0,52
13	0,55	0,58	0,52	0,52
12	0,56	0,55	0,51	0,52
11	0,56	0,55	0,51	0,53
10	0,54	0,55	0,53	0,52
9	0,52	0,53	0,55	0,53
8	0,51	0,52	0,54	0,53
7	0,51	0,57	0,52	0,52
6	0,51	0,55	0,51	0,52
5	0,51	0,54	0,53	0,52
4	0,52	0,53	0,52	0,53
3	0,49	0,55	0,53	0,53
2	0,49	0,54	0,52	0,51
1	0,49	0,53	0,52	0,51

Table 6: Results all hidden layers model finetuned with 8K FTE.

Layer	LR MNLI	LR HANS	CCS MNLI	CCS HANS
24	0,85	0,67	0,61	0,52
23	0,86	0,72	0,65	0,51
22	0,85	0,69	0,59	0,52
21	0,84	0,57	0,81	0,52
20	0,77	0,53	0,79	0,51
19	0,61	0,57	0,52	0,53
18	0,58	0,53	0,53	0,52
17	0,57	0,54	0,53	0,52
16	0,57	0,53	0,52	0,52
15	0,61	0,57	0,53	0,51
14	0,61	0,57	0,51	0,52
13	0,61	0,58	0,53	0,52
12	0,60	0,57	0,53	0,51
11	0,59	0,57	0,51	0,52
10	0,57	0,55	0,52	0,52
9	0,53	0,59	0,51	0,51
8	0,52	0,55	0,52	0,52
7	0,51	0,56	0,52	0,52
6	0,50	0,55	0,52	0,51
5	0,50	0,56	0,52	0,52
4	0,50	0,54	0,52	0,52
3	0,51	0,55	0,52	0,52
2	0,50	0,55	0,51	0,53
1	0,50	0,54	0,51	0,52

Table 7: Results all hidden layers model finetuned with 16K FTE.