Identifying and Mitigating Spurious Correlations for Improving Robustness in NLP Models

Tianlu Wang
University of Virginia
tianlu@virginia.edu

Rohit Sridhar
Georgia Institute of Technology
rohitsridhar@gatech.edu

Diyi Yang
Georgia Institute of Technology
dyang888@gatech.edu

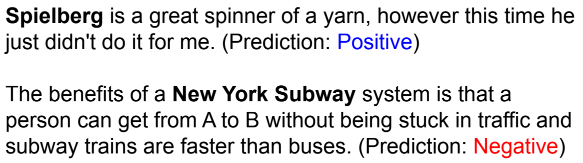
Xuezhi Wang
Google Research
xuezhiw@google.com

Abstract

Recently, NLP models have achieved remarkable progress across a variety of tasks; however, they have also been criticized for being not robust. Many robustness problems can be attributed to models exploiting *spurious correlations*, or *shortcuts* between the training data and the task labels. Most existing work identifies a limited set of task-specific shortcuts via human priors or error analyses, which requires extensive expertise and efforts. In this paper, we aim to automatically identify such spurious correlations in NLP models at scale. We first leverage existing interpretability methods to extract tokens that significantly affect model's decision process from the input text. We then distinguish "genuine" tokens and "spurious" tokens by analyzing model predictions across multiple corpora and further verify them through knowledge-aware perturbations. We show that our proposed method can effectively and efficiently identify a scalable set of "shortcuts", and mitigating these leads to more robust models in multiple applications.

1 Introduction

Despite great progress has been made over improved accuracy, deep learning models are known to be brittle to out-of-domain data (Hendrycks et al., 2020; Wang et al., 2019), adversarial attacks (McCoy et al., 2019; Jia and Liang, 2017; Jin et al., 2020), partly due to sometimes the models have exploited *spurious correlations* in the existing training data (Tu et al., 2020; Sagawa et al., 2020). In Figure 1, we show an example of a sentiment classification model making spurious correlations over the phrases "*Spielberg*" and "*New York Subway*" due to their high co-occurrences with positive and negative labels respectively in the training data.



Spielberg is a great spinner of a yarn, however this time he just didn't do it for me. (Prediction: Positive)

The benefits of a **New York Subway** system is that a person can get from A to B without being stuck in traffic and subway trains are faster than buses. (Prediction: Negative)

Figure 1: Examples of spurious correlations in sentiment classification task. A sentiment classification model takes *Spielberg* and *New York Subway* as shortcuts and makes wrong predictions.

Most existing work quantifies spurious correlations in NLP models via a set of pre-defined patterns based on human priors and error analyses over the models, e.g., syntactic heuristics for Natural Language Inference (McCoy et al., 2019), synonym substitutions (Alzantot et al., 2018), or adding adversarial sentences for QA (Jia and Liang, 2017). More recent work on testing models' behaviour using CheckList (Ribeiro et al., 2020) also used a pre-defined series of test types, e.g., adding negation, temporal change, and switching locations/person names. However, for safe deployment of NLP models in the real world, in addition to predefining a small or limited set of patterns which the model could be vulnerable to, it is also important to proactively *discover* and *identify* models' unrobust regions automatically and comprehensively.

In this work, we introduce a framework to automatically identify spurious correlations exploited by the model, sometimes also denoted as "shortcuts" in prior work (Geirhos et al., 2020; Minderer et al., 2020)[1], at a large scale. Our proposed framework differs from existing literature with a focus more on automatic shortcut identification, instead of pre-defining a limited set of shortcuts or learning from human annotations (Table 1). Our framework works as follows: given a task and a trained model, we first utilize interpretability methods, e.g., attention scores (Clark et al., 2019b; Kovaleva et al., 2019) and integrated gradient (Sundararajan et al., 2017) which are commonly used for interpreting model's decisions, to automatically extract tokens that the model deems as important for task label

[1] Throughout the paper we use spurious correlations and shortcuts interchangeably.

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1719 - 1729
July 10-15, 2022 ©2022 Association for Computational Linguistics

	Objective	Approach for shortcut identification
He et al. (2019)	Robustness against *known* shortcuts	Pre-defined
Clark et al. (2019a)	Robustness against *known* shortcuts	Pre-defined
Clark et al. (2020)	Robustness against *unknown* shortcuts	A low-capacity model to specifically learn shortcuts
Wang and Culotta (2020a)	Identify *unknown* shortcuts for robustness	A classifier over human annotated examples
This paper	Identify *unknown* shortcuts for robustness	Automatic identification with interpretability methods

Table 1: Comparison of our work and other related literature.

prediction. We then introduce two extra steps to further categorize the extracted tokens to be "genuine" or "spurious". We utilize a *cross-dataset* analysis to identify tokens that are more likely to be "shortcut". The intuition is that if we have data from multiple domains for the same task, then "genuine" tokens are more likely to remain useful to labels across domains, while "spurious" tokens would be less useful. Our last step further applies a *knowledge-aware perturbation* to check how stable the model's prediction is by perturbing the extracted tokens to their semantically similar neighbors. The intuition is that a model's prediction is more likely to change when a "spurious" token is replaced by its semantically similar variations. To mitigate these identified "shortcuts", we propose a simple yet effective *targeted* mitigation approach to prevent the model from using those "shortcuts" and show that the resulting model can be more robust. Our code and data have been made publicly.[2] Our contributions are as follows:

- We introduce a framework to automatically identify shortcuts in NLP models at scale. It first extracts important tokens using interpretability methods, then we propose *cross-dataset analysis* and *knowledge-aware perturbation* to distinguish spurious correlations from genuine ones.
- We perform experiments over several benchmark datasets and NLP tasks including sentiment classification and occupation classification, and show that our framework is able to identify more subtle and diverse spurious correlations. We present results showing the identified shortcuts can be utilized to improve robustness in multiple applications, including better accuracy over challenging datasets, better adaptation across multiple domains, and better fairness implications over certain tasks.

2 Related Work

Interpretability There has been a lot of work on better interpreting models' decision process, e.g., understanding BERT (Clark et al., 2019b; Kovaleva et al., 2019) and attention in transformers (Hao et al., 2020), or through text generation models (Narang et al., 2020). In this paper we utilize the attention scores as a generic way to understand what features a model relies on for making its predictions. Other common model interpretation techniques (Sundararajan et al., 2017; Ribeiro et al., 2016), or more recent work on hierarchical attentions (Chen et al., 2020) and contrastive explanations (Jacovi et al., 2021), can be used as well. In Pruthi et al. (2020), the authors found that attention scores can be manipulated to deceive human decision makers. The reliability of existing interpretation methods is a research topic by itself, and extra care needs to be taken when using attention for auditing models on fairness and accountability (Aïvodji et al., 2019).

Robustness and Bias An increasing body of work has been conducted on understanding robustness in deep neural networks, particularly, how models sometimes might exploit spurious correlations (Tu et al., 2020; Sagawa et al., 2020) and take shortcuts (Geirhos et al., 2020), leading to vulnerability in generalization to out-of-distribution data or adversarial examples in various NLP tasks such as NLI (McCoy et al., 2019), Question-Answering (Jia and Liang, 2017), and Neural Machine Translation (Niu et al., 2020). Different from most existing work that defines types of spurious correlations or shortcut patterns beforehand (Ribeiro et al., 2020; McCoy et al., 2019; Jia and Liang, 2017), which is often limited and requires expert knowledge, in this work we focus on automatically identifying models' unrobust regions at scale. Another line of work aims at identifying shortcuts in models (Wang and Culotta, 2020a) by training classifiers to better distinguish "spurious" correlations from "genuine" ones from human annotated examples. In contrast, we propose a cross-dataset approach and a knowledge-aware perturbation approach to automate this identification process with less human intervention in-between.

[2] https://github.com/tianlu-wang/Identifying-and-Mitigating-Spurious-Correlations-for-Improving-Robustness-in-NLP-Models

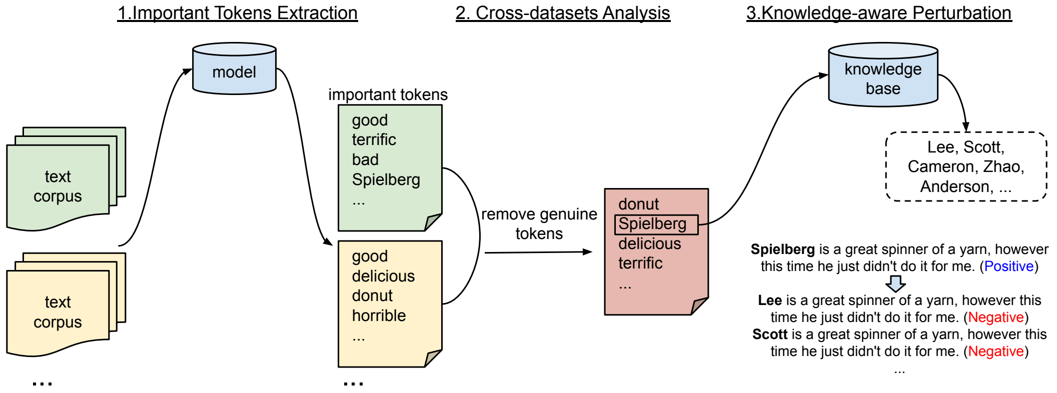

1.Important Tokens Extraction
2. Cross-datasets Analysis
3.Knowledge-aware Perturbation
text corpus
text corpus
...
model
important tokens
good
terrific
bad
Spielberg
...
good
delicious
donut
horrible
...
...
remove genuine tokens
donut
Spielberg
delicious
terrific
...
knowledge base
Lee, Scott, Cameron, Zhao, Anderson, ...
Spielberg is a great spinner of a yarn, however this time he just didn't do it for me. (Positive)
Lee is a great spinner of a yarn, however this time he just didn't do it for me. (Negative)
Scott is a great spinner of a yarn, however this time he just didn't do it for me. (Negative)
...

Figure 2: Our proposed pipeline to identify spurious correlations at scale. In the first step, we extract important tokens from input text. In the second step, we analyze extracted tokens from various datasets to identify likely "spurious" tokens. Finally, we further validate the output from the second step through knowledge-aware perturbation.

Mitigation Multiple approaches have been proposed to mitigate shortcut learning and data biases (Clark et al., 2020; Bras et al., 2020; Zhou and Bansal, 2020; Minderer et al., 2020), through data augmentation (Jin et al., 2020; Alzantot et al., 2018), domain adaptation (Blitzer et al., 2006, 2007), and multi-task learning (Tu et al., 2020). Du et al. (2021) proposes to mitigate shortcuts by suppressing model's prediction on examples with a large shortcut degree. Recent study has also shown removing spurious correlations can sometimes hurt model's accuracy (Khani and Liang, 2021). Orthogonal to existing works, we propose to first identify unrobust correlations in an NLP model and then propose a *targeted* mitigation to encourage the model to rely less on those unrobust correlations.

3 Framework for Identifying Shortcuts

In this section, we introduce our framework to identify spurious correlations in NLP models. Our overall framework consists of first identifying tokens important for models' decision process, followed by a cross-dataset analysis and a knowledge-aware perturbation step to identify spurious correlations.

3.1 Identify Tokens Key to Model's Decision

The first step of the framework aims to identify the top-K most important tokens that affect model's decision making process. We look at the importance at the token-level.[3] In general, depending on how the tokens are being used in model's decision process, they can be roughly divided into three categories: "genuine", "spurious", and others (e.g., tokens that are not useful for a model's prediction). *Genuine* tokens are tokens that causally affect a task's label (Srivastava et al., 2020; Wang and Culotta, 2020b), and thus the correlations between those tokens and the labels are what we expect the model to capture and to more heavily rely on. On the other hand, *spurious* tokens, or *shortcuts* as commonly denoted in prior work (Geirhos et al., 2020; Minderer et al., 2020), are features that correlate with task labels but are not genuine, and thus might fail to transfer to challenging test conditions (Geirhos et al., 2020) or out-of-distribution data; spurious tokens do not causally affect task labels (Srivastava et al., 2020; Wang and Culotta, 2020b).

In this step, we will extract both genuine tokens and shortcut tokens because they are both likely to affect a model's prediction. We rely on interpretability techniques to collect information on whether a certain input token is important to model's decision making. In this paper, we use the attention score in BERT-based models as an explanation of model predictions (Clark et al., 2019b; Kovaleva et al., 2019), due to its simplicity and fast computation. Recent work (Jiaao et al., 2021) also reveals that attention scores outperform other explanation techniques in regularizing redundant information. Other techniques (Ribeiro et al., 2016; Sundararajan et al., 2017; Chen et al., 2020; Jacovi et al., 2021) can also be used in this step. As an example, given a sentence "*Spielberg is a good director.*", assuming "*good*" is a genuine token and "*Spielberg*" is a shortcut token, we expect that in a BERT-based sentiment classification model, the attention scores for "*good*" and "*Spielberg*" are

[3] In this paper, we mostly focus on unigrams. Our method can also be easily extended to multi-gram, text span or other type of features by summing the attention scores over spans. For a vocabulary of wordpieces as used in BERT, we concatenate wordpieces with a prefix of "##" to form unigrams and sum the attention scores.

higher and thus will be extracted as important tokens. On the other hand, for "*is*", "*a*" and "*director*" the attention scores would be lower as they are relatively less useful to the model decision.

We now describe this step using sentiment classification task as an example (more details can be found in Algorithm 1). Let f be a well trained sentiment classification model. Given a corpus $\mathcal{D}$, for each input sentence s_i, $i = 1, \ldots, n$ for a total of n sentences in the corpus, we apply f on it to obtain the output probability p_i^{pos} and p_i^{neg} for positive and negative label respectively. We then extract attention scores $\{a_i^1, a_i^2, \ldots, a_i^m\}$ for tokens $\{t_i^1, t_i^2, \ldots, t_i^m\}$ in sentence s_i, where m is the length of the sentence. In BERT-based classification models, the embedding of [CLS] token in the final layer is fed to a classification layer. We thus extract the attention scores of each token t used for computing the embedding of the [CLS] token and average them across different heads. If $p_i^{pos} > p_i^{neg}$, we obtain the updated attention score $\tilde{a}_i^j = a_i^j * p_i^{pos}$, otherwise $\tilde{a}_i^j = -a_i^j * p_i^{neg}$. For each token t in the vocabulary $\mathcal{V}$, we compute the average attention score: $\bar{a}_t = \frac{1}{mn} \cdot \Sigma_{i=1}^{n} \Sigma_{j=1}^{m} [\tilde{a}_i^j \cdot \mathbb{1}(t_i^j = t)]$, where we aggregate the attention scores $\tilde{a}_i^j$ for token t, across all n sentences in the corpus. We then normalize the attention scores across the vocabulary to obtain the importance score for each token t: $I_t = \bar{a}_t / \Sigma_{t \in \mathcal{V}} \bar{a}_t$. This can lead to very small I_t for certain tokens, thus we take the log of all importance scores to avoid underflow, $I'_t = \log(I_t)$.

So far, we have computed the importance score for each token. However, we observe that some tokens appearing only very a few times could accidentally have very high importance scores. Thus, we propose to penalize the tokens with low frequencies: $\hat{I}_t = I'_t - \lambda / \log(1 + c_t)$, where c_t is the frequency of token t and λ is a temperature parameter to adjust the degree that we want to penalize over the frequency.

3.2 Cross-Dataset Stability Analysis

As mentioned before, the tokens that are important to a model's prediction could be either genuine or spurious, thus in this step, we want to categorize the extracted tokens into these two categories and maintain a list of tokens that are more likely to be "spurious".

```
Algorithm 1: Important Token Extraction.
Input  : Sentiment classification model: $f$
         Text corpus: $\mathcal{D}$
// Obtain attention scores for tokens in each
   input sentence $s_i \in \mathcal{D}$:
for $i = 1$ to $n$ do
    $p_i^{pos}, p_i^{neg}, \{a_i^1, a_i^2, ..., a_i^m\} = f(s_i)$;
    for $j = 1$ to $m$ do
        if $p_i^{pos} > p_i^{neg}$: $\tilde{a}_i^j = a_i^j \cdot p_i^{pos}$;
        else: $\tilde{a}_i^j = -a_i^j \cdot p_i^{neg}$;
    end
end
// Use $\{\tilde{a}_i^j\}$ to compute an importance score
   for each token $t$ in the vocabulary $\mathcal{V}$:
$Importance = dict()$
for $i = 1$ to $n$ do
    for $j = 1$ to $m$ do
        $Importance[t_i^j]$.append($\tilde{a}_i^j$);
    end
end
// Normalize the importance score and
   penalize low-frequency tokens:
for $t$ in $\mathcal{V}$ do
    $\bar{a}_t = average(Importance[t])$;
    $I_t = \bar{a}_t / \Sigma_{t \in \mathcal{V}} \bar{a}_t$;
    $I'_t = \log(I_t)$;
    $\hat{I}_t = I'_t - \lambda / \log(1 + frequency[t])$;
end
Output : A list of tokens sorted according to
         their importance scores:
         $\{t_1, t_2, ..., t_{|\mathcal{V}|}\}$,
         where $\hat{I}_{t_i} \geq \hat{I}_{t_2} \geq ... \geq \hat{I}_{t_{|\mathcal{V}|}}$
```

In many real-world NLP tasks, if we have access to datasets from different sources or domains, then we can perform a *cross-dataset* analysis to more effectively identify "spurious" tokens. The reasoning is that "spurious" tokens tend to be important for a model's decision making on one dataset but are less likely to transfer or generalize to other datasets, e.g. "*Spielberg*" could be an important token for movie reviews but is not likely to be useful on other review datasets (e.g., for restaurants or hotels). On the other hand, genuine tokens are more likely to be important across multiple datasets, for example, tokens like "*good*", "*bad*", "*great*", "*terrible*" should remain useful across various sentiment classification datasets. Thus, in this step, we try to distinguish "genuine" tokens from "spurious" tokens from the top extracted important tokens after the first step. Our idea is to compare tokens' im-

Shortcut token: **bread** *Original:* I bought this in the hopes it would keep **bread** I made fresh. However, after a few times of usings the I found out that moister w still getting in bread would become stale or moldy ...(Neg) *Perturbed:* I bought this in the hopes it would keep **loaf** I made fresh. However, after a few times of usings the I found out that moister w still getting in bread would become stale or moldy ... (Pos)
Shortcut token: **iPhone** *Original:* I lost my original TV remote, and found this one thinking it was the same one. ... Now this one is merely a back up. Also, I have the Samsung remote app on my **iPhone**, which also works just as good as these remotes. (Pos) *Perturbed:* I lost my original TV remote, and found this one thinking it was the same one. ... Now this one is merely a back up. Also, I have the Samsung remote app on my **ipod**, which also works just as good as these remotes. (Neg)

Table 2: Examples of shortcut tokens with significant performance drop during knowledge-aware perturbation.

portance ranking and find the ones that have very different ranks across datasets.

To this end, we conduct a cross-dataset stability analysis. Specifically, we apply the same model f on two datasets A and B, and obtain two importance ranking lists. Since importance scores may have different ranges on the two datasets, we normalize all importance scores to adjust the value to be in the range of $[0, 1]$:

$$\tilde{I}_t^A = \frac{\hat{I}_t^A - \min(\{\hat{I}_t^A | t \in \mathcal{V}\})}{\max(\{\hat{I}_t^A | t \in \mathcal{V}\}) - \min(\{\hat{I}_t^A | t \in \mathcal{V}\})}$$

$$\tilde{I}_t^B = \frac{\hat{I}_t^B - \min(\{\hat{I}_t^B | t \in \mathcal{V}\})}{\max(\{\hat{I}_t^B | t \in \mathcal{V}\}) - \min(\{\hat{I}_t^B | t \in \mathcal{V}\})}$$

where $\tilde{I}_t^A$ and $\tilde{I}_t^B$ are normalized importance scores on dataset A and B respectively. We then subtract $\tilde{I}_t^B$ from $\tilde{I}_t^A$ and re-rank all tokens according to their differences. Tokens with largest differences are the ones with high importance scores in dataset A but low importance scores in dataset B, thus they are more likely to be "shortcut" tokens in dataset A. Similarly, we can also extract tokens with largest differences from dataset B by subtract $\tilde{I}_t^A$ from $\tilde{I}_t^B$.

3.3 Knowledge-aware Perturbation

The cross-dataset analysis is an efficient way to remove important tokens that are "genuine" across multiple datasets, after which we can obtain a list with tokens that are more likely to be "spurious". However, on this list, domain-specific genuine tokens can still be ranked very high, e.g., "*ambitious*" from a movie review dataset and "*delicious*" from a restaurant review dataset. This is because domain-specific genuine tokens have similar characteristics as shortcuts, they are effective for a model's decision making on a certain dataset but could appear very rarely (and thus could be deemed as not important) on another dataset. Hence, in this section, we further propose a slightly more expensive and a more fine-grained approach to verify whether a token is indeed "spurious", through *knowledge-aware perturbation*.

For each potential shortcut token, we extract N synonyms by leveraging the word embeddings curated for synonym extraction (Mrkšić et al., 2016), plus WordNet (Miller, 1995) and DBpedia (Auer et al., 2007). More specifically, for each top token t in the list generated by the previous step, we first search counter-fitting word vectors to find synonyms with cosine similarity larger than a threshold[4] τ. Additionally we search in WordNet and DBpedia to obtain a maximum of N synonyms for each token t. Then we extract a subset S_t from $\mathcal{D}$, which consists of sentences containing t. We perturb all sentences in S_t by replacing t with its synonyms. The resulted perturbed set S_t' is N times of the original set S_t. We apply model f on S_t and S_t' and obtain accuracy acc_t and acc_t'. Since we only perturb S_t with t's synonyms, the semantic meaning of perturbed sentences should stay close to the original sentences. Thus, if t is a genuine token, acc_t' is expected to be close to acc_t. On the other hand, if t is a shortcut, model prediction can be different even the semantic meaning of the sentence does not change a lot (see examples in Table 2). Thus, we assume tokens with larger differences between acc_t and acc_t' are more likely to be shortcuts and tokens with smaller differences are more likely to be domain specific "genuine" words. From the potential shortcut token list computed in Sec 3.2, we remove tokens with performance difference smaller than δ to further filter domain specific "geniue" tokens .

3.4 Mitigation via Identified Shortcuts

In this section, we describe how the identified shortcuts can be further utilized to improve robustness in NLP models. More specifically, we propose *targeted* approaches to mitigate the identified

[4] We set it as 0.5 following the set up in (Jin et al., 2020).

Dataset	Top **important** tokens extracted from each dataset
SST-2	terrific, impeccable, exhilarating, refreshingly, irresistible, heartfelt, thought-provoking, ...
Yelp	Awesome, Definitely, is, Excellent, Very, Great, Good, Best, attentive, worth, definitely, Highly, ...
Amazon Kitchen	utensils, thermometer, Cuisinart, Definitely, Pyrex, Bought, utensil, Arrived, Recommend, ...
Dataset	Top **shortcuts** extracted from each dataset (verified by human annotators)
SST-2	recycled, seal, sitcom, longest, fallen, qualities, rises, impact, translate, emphasizes, ...
Yelp	ambiance, tastes, bartenders, patio, burgers, staff, watering, donuts, cannot, pancakes, regulars, ...
Amazon Kitchen	utensils, Cuisinart, Rachael, Pyrex, utensil, Breville, Zojirushi, Corelle, Oxo, dehumidifier, ...

Table 3: Top *important* tokens and top *shortcut* tokens identified by our proposed framework and further verified by human annotators. Many shortcuts reflect the characteristics of the datasets, e.g. "captures" from a movie review dataset, "burgers" from a restaurant review dataset and brand names from an Amazon kitchen review dataset.

Dataset	Method	@10		@20		@50	
		Prec.	Impor.	Prec.	Impor.	Prec.	Impor.
SST-2	1	0.00	-	0.05	0.97	0.02	0.96
	2	0.10	0.95	0.05	0.94	0.04	0.93
	3	0.40	0.90	0.35	0.87	0.32	0.85
Yelp	1	0.10	0.96	0.05	0.95	0.18	0.95
	2	0.40	0.89	0.25	0.89	0.30	0.88
	3	0.60	0.89	0.50	0.87	0.56	0.87
Amazon Kitchen	1	0.70	0.98	0.80	0.96	0.78	0.95
	2	1.00	0.97	1.00	0.95	1.00	0.95
	3	1.00	0.97	1.00	0.95	1.00	0.95

Table 4: We report the precision as well as the averaged importance score $\tilde{I}$ of identified "shortcuts" after each step based on our framework. The identified "shortcut" is a true shortcut or not is verified by 3 independent human annotators (Amazon Turkers). We can see that the precision increases after each step in our framework, demonstrating the utility of cross-dataset analysis (step 2) and knowledge-aware perturbation (step 3).

shortcuts including three variants: (1) a *training-time* mitigation approach: we mask out the identified shortcuts during training time and re-train the model; (2) an *inference-time* mitigation approach: we mask out the identified shortcuts during inference time only, in this way we save the extra cost of re-training a model; (3) we combine both approach (1) and (2). In the experiment section, we will demonstrate the effect of each approach over a set of benchmark datasets. We found that by masking out shortcuts in datasets, models generalize better to challenging datasets, out-of-distribution data, and also become more fair.

4 Experiments

4.1 Tasks and Datasets

Task 1: Sentiment classification. For the task of sentiment classification, we use several datasets in our experiments. To find shortcuts in Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) dataset, we first train a model on SST-2 training set which consists of $67,349$ sentences. We then evaluate the model on SST-2 training set[5] and Yelp (Asghar, 2016) test set and obtain attention scores. For cross-dataset analysis, we compare the important tokens extracted from SST-2 and Yelp. Similarly, we train another model on $80,000$ amazon kitchen reviews (He and McAuley, 2016), and apply it on the kitchen review dev set and the amazon electronics dev set, both having $10,000$ reviews.

Task 2: Occupation classification. Following Pruthi et al. (2020), we use the biographies (De-Arteaga et al., 2019) to predict whether the occupation is a surgeon or physician (non-surgeon). The training data consists of $17,629$ biographies and the dev set contains $2,519$ samples.

Models. We use the attention scores over BERT (Devlin et al., 2019) based classification models as they have achieved the state-of-art performance. Note that our proposed framework can also be easily extended to models with different architectures. BERT-based models have the advantage that we can directly use the attention scores as explanations of model decisions. For models with other architectures, we can use explanation techniques such as LIME (Ribeiro et al., 2016) or Path Integrated Gradient approaches (Sundararajan et al., 2017) to provide explanations.

[5] We use training set of SST-2 because the test set has a very limited number of examples.

Evaluation. Evaluating identified shortcuts in machine learning or deep leaning based models can be difficult. We do not have ground-truth labels for the shortcuts identified through our framework, and whether a token is a shortcut or not can be subjective even with human annotators, and it can further depend on the context. Faced with these challenges, we carefully designed a task and adopted Amazon Mechanical Turk for evaluation. We post the identified shortcuts after each step in our framework, along with several sample sentences containing the token, as additional context, to the human annotator. We ask the question "does the word determine the sentiment in the sentence" and ask the annotator to provide a "yes"/"no" answer[6] to the question based on the answer that holds true for the majority of the provided sentences (we also experimented with adding an option of "unsure" but found most annotators do not choose that option). Each identified shortcut is verified by 3 annotators.

4.2 Experimental Results

We summarized the top important tokens after each step in our framework (Table 3). We also report the precision score (the percentage of tokens) out of the top 50 tokens identified as true shortcuts by human annotators in Table 4.

Across all datasets, we see that the precision score increases after each step, which demonstrates that our proposed framework can consistently improve shortcut identification more precisely. Specifically, after the first step, the precision score of shortcuts is low[7] because most of the top extracted tokens are important tokens only (thus many of them are genuine). After the second step (cross-dataset analysis) and the third step (knowledge-aware perturbation), we see a significant increase of the shortcuts among the top-K extracted tokens. Table 2 shows examples of perturbing shortcut tokens leading to model predictions changes.

Agreement analysis over annotations. Since this annotation task is non-trivial and sometimes subjective, we further compute the intraclass correlation score (Bartko, 1966) for the Amazon Mechanical Turk annotations. Our collected annotations reaches an intraclass correlation score of 0.72, showing a good agreement among annotators. Another agreement we analyze is showing annotators 5 sample sentences compared to showing them all sentences, to avoid sample bias. We ask annotators to annotate a batch of 25 tokens with all sentences containing the corresponding token shown to them. The agreement reaches 84.0%, indicating that showing 5 sample sentences does not significantly affect annotator's decision on the target token. More details of Amazon Mechanical Turk interface can be found in the Appendix.

4.3 A Case Study: Occupation Classification

Pruthi et al. (2020) derived an occupation dataset to study the gender bias in NLP classification tasks. The task is framed as a binary classification task to distinguish between "surgeons" and "physicians". These two occupations are chosen because they share similar words in their biographies and a majority of surgeons are male. The dataset is further tuned – downsample minority classes (female surgeons and male physicians) by a factor of ten to encourage the model to rely on gendered words to make predictions. Pruthi et al. (2020) also provides a pre-specified list of impermissible tokens [8] that a robust model should assign low attention scores to. We instead treat this list of tokens as shortcuts and analyze the efficacy of our proposed framework on identifying these tokens. These impermissible tokens can be regarded as shortcuts because they only reflect the gender of the person, thus by definition should not affect the decision of a occupation classification model. Table 6 presents the result on identifying the list of impermissible tokens. Among the top ten tokens selected by our method, 6 of them are shortcuts. Furthermore, 9 out of 12 impermissible tokens are captured in the top 50 tokens selected by our method. This further demonstrates that our method can effectively find shortcuts in this occupation classification task, in a more automated way compared to existing approaches that rely on pre-defined lists.

4.4 Mitigating Shortcuts

We also study mitigating shortcuts by masking out the identified shortcuts. Specifically, we use shortcut tokens identified by human annotators and mask them out in training set and re-train the model (Train RM), during test time directly (Test

[6] In the instruction, we further specify "select yes" if the highlighted word is a determining factor for the sentiment label, and we provide a few example sentences along with their shortcuts as references. The exact template is shown in the Appendix.

[7] Some cells have "-" importance score due to no shortcut is identified by human annotators in the top-K identified tokens.

[8] he, she, her, his, him, himself, herself, mr, ms, mr., mrs., ms. We removed "hers" and "mrs" from the original list since they do not appear in dev data.

Methods	SST-2 → Kitchen	SST-2 → Electronics	Kitchen → SST-2	Kitchen → Electronics	Electronics → SST-2	Electronics → Kitchen
No Mitigation	87.43	**84.30**	71.45	98.22	73.05	98.79
Test RM	87.50	83.96	71.56	98.07	72.94	98.77
Train RM	87.72	84.13	**72.82**	**98.60**	**74.08**	98.79
Train & Test RM	**87.76**	83.74	**72.82**	**98.62**	**74.08**	**98.80**

Table 5: Domain generalization results on SST-2 and Amazon Kitchen/Electronics datasets. RM means shortcuts removed, Train/Test corresponds to shortcuts removal during training and test time, respectively.

Top 10 extracted tokens	Precision	Recall
ms. , mrs. , she , her , he , reviews, been, favorite, his , practices	0.60	0.50

Table 6: Identified shortcuts (highlighted tokens are overlapped with the pre-specified impermissible tokens from Pruthi et al. (2020)) in occupation classification.

Dataset	Methods	C1	C2
Amazon Kitchen	No Mitigation	99.15	0.0
	Test RM	99.21	0.18
	Train RM	99.15	0.24
	Train & Test RM	99.15	0.24

Table 7: Accuracy on challenging datasets. C1: test subset that has shortcuts; C2: test subset that has shortcuts and are wrongly predicted by the original model.

	Male	Female	Δ	Overall
No Mitigation	94.02	99.50	5.48	97.46
Test RM	92.28	96.36	4.08	94.84
Train RM	93.26	92.40	0.86	92.66
Train & Test RM	94.46	99.06	4.60	97.34

Table 8: Accuracy and performance gap of male and female groups in Occupation Classification task.

RM), and both (Train & Test RM) as described in Sec 3.4. We evaluate these three approaches in multiple settings: 1) domain generalization; 2) challenging datasets; 3) gender bias. As shown in Table 5, masking out shortcuts, especially in training data, can improve model's generalization to out-of-distribution data. Note in this setting, different from existing domain transfer work (Pan and Yang, 2010), we do not assume access to labeled data in the target domain during training, instead we use our proposed approach to identify potential shortcuts that can generalize to unseen target domains. As a result, we also observe model's performance improvement on challenging datasets (Table 7). Table 8 demonstrates that mitigating shortcuts helps to reduce the performance gap (Δ) between male and female groups, resulting in a fairer model. Note the original performance might degrade slightly due to models learning different but more robust feature representations, consistent with findings in existing work (Tsipras et al., 2019).

λ	4	6	8	10
4	1.00	0.78	0.62	0.56
6	0.78	1.00	0.84	0.76
8	0.62	0.84	1.00	0.92
10	0.56	0.76	0.92	1.00

Table 9: Overlap of top 50 tokens when changing λ.

Dataset	Method	@10 Prec.	@20 Prec.	@50 Prec.
SST-2	Attention	0.40	0.35	0.32
	Integrated Gradient	0.30	0.3	0.34
Yelp	Attention	0.60	0.50	0.56
	Integrated Gradient	0.50	0.55	0.60

Table 10: Ablation study on using Integrated Gradient to extract important tokens.

Ablation Study We conduct an ablation study of changing the hyper-parameter λ in the first step of extracting important tokens. As shown in Table 9, our method is not very sensitive to the choice of λ. In Table 10, we show that Attention scores and Integrated Gradient can both serve as a reasonable method for extracting important tokens in our first step, suggesting the flexibility of our framework.

5 Conclusion

In this paper, we aim to improve NLP models' robustness via identifying spurious correlations automatically at scale, and encouraging the model to rely less on those identified shortcuts. We perform experiments and human studies over several benchmark datasets and NLP tasks to show a scalable set of shortcuts can be efficiently identified through our framework. Note that we use existing interpretability approaches as a proxy to better understand how a model reaches its prediction, but as pointed out by prior work, the interpretability methods might not be accurate enough to reflect how a model works (or sometimes they could even deceive human decision makers). We acknowledge this as a limitation, and urge future research to dig deeper and develop better automated methods with less human intervention or expert knowledge in improving models' robustness.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. 2019. Fairwashing: the risk of rationalization. In *Proceedings of the 34th International Conference on Machine Learning*.

John J Bartko. 1966. The intraclass correlation coefficient as a measure of reliability. *Psychological reports*, 19(1):3–11.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia. Association for Computational Linguistics.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5578–5593, Online. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019a. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019b. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios. *Proceedings of the Conference on Fairness, Accountability, and Transparency*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *NAACL 2021*.

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.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Self-attention attribution: Interpreting information interactions inside transformer. In *AAAI 2021*.

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

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16.

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

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

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

Chen Jiaao, Shen Dinghan, Chen Weizhu, and Yang Diyi. 2021. Hiddencut: Simple data augmentation for natural language understanding with better generalizability. In *Proceedings of the 59th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Zhou, and Peter Szolovits. 2020. Is BERT really robust? Natural language attack on text classification and entailment. In *AAAI*.

Fereshte Khani and Percy Liang. 2021. Removing spurious features can hurt accuracy and affect groups disproportionately. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 196–205, New York, NY, USA. Association for Computing Machinery.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. 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.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. 2020. Automatic shortcut removal for self-supervised representation learning. In *Proceedings of the 37th International Conference on Machine Learning*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of HLT-NAACL*.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *CoRR*, abs/2004.14546.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544, Online. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

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

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. 2020. Robustness to spurious correlations via human annotations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9109–9119. PMLR.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

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

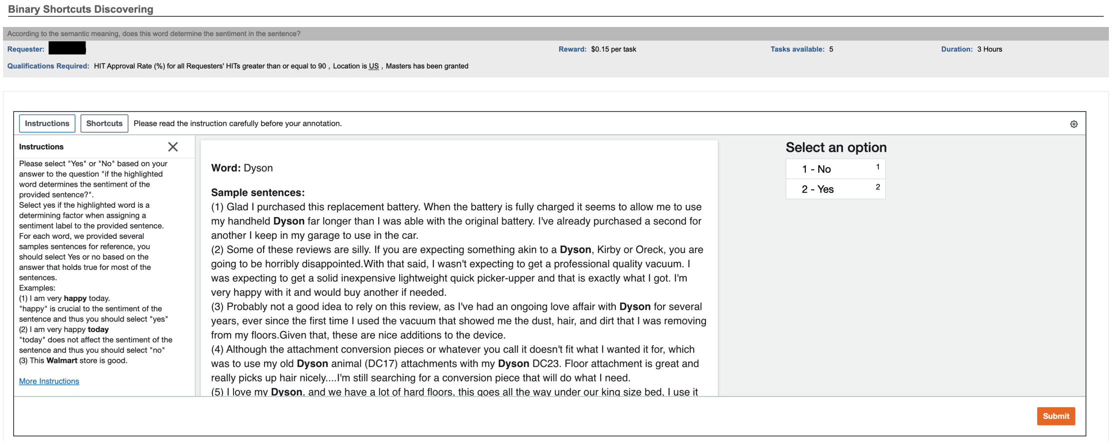
Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019. Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Zhao Wang and Aron Culotta. 2020a. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

Zhao Wang and Aron Culotta. 2020b. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *AAAI*.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online. Association for Computational Linguistics.

A Appendix



Binary Shortcuts Discovering

According to the semantic meaning, does this word determine the sentiment in the sentence?

Requester: [redacted]   Reward: $0.15 per task   Tasks available: 5   Duration: 3 Hours

Qualifications Required: HIT Approval Rate (%) for all Requesters' HITs greater than or equal to 90 , Location is US , Masters has been granted

Please read the instruction carefully before your annotation.

Instructions

Please select "Yes" or "No" based on your answer to the question "if the highlighted word determines the sentiment of the provided sentence?".
Select yes if the highlighted word is a determining factor when assigning a sentiment label to the provided sentence.
For each word, we provided several samples sentences for reference, you should select Yes or no based on the answer that holds true for most of the sentences.
Examples:
(1) I am very **happy** today.
"happy" is crucial to the sentiment of the sentence and thus you should select "yes"
(2) I am very happy **today**
"today" does not affect the sentiment of the sentence and thus you should select "no"
(3) This **Walmart** store is good.

Word: Dyson

Sample sentences:
(1) Glad I purchased this replacement battery. When the battery is fully charged it seems to allow me to use my handheld **Dyson** far longer than I was able with the original battery. I've already purchased a second for another I keep in my garage to use in the car.
(2) Some of these reviews are silly. If you are expecting something akin to a **Dyson**, Kirby or Oreck, you are going to be horribly disappointed.With that said, I wasn't expecting to get a professional quality vacuum. I was expecting to get a solid inexpensive lightweight quick picker-upper and that is exactly what I got. I'm very happy with it and would buy another if needed.
(3) Probably not a good idea to rely on this review, as I've had an ongoing love affair with **Dyson** for several years, ever since the first time I used the vacuum that showed me the dust, hair, and dirt that I was removing from my floors.Given that, these are nice additions to the device.
(4) Although the attachment conversion pieces or whatever you call it doesn't fit what I wanted it for, which was to use my old **Dyson** animal (DC17) attachments with my **Dyson** DC23. Floor attachment is great and really picks up hair nicely....I'm still searching for a conversion piece that will do what I need.
(5) I love my **Dyson**, and we have a lot of hard floors, this goes all the way under our king size bed, I use it

Select an option

1 - No 1
2 - Yes 2

Figure 3: Amazon mechanical Turk interface.