Finding Dataset Shortcuts with Grammar Induction

Dan Friedman, Alexander Wettig, Danqi Chen
Department of Computer Science, Princeton University
{dfriedman,awettig,danqic}@cs.princeton.edu

Abstract

Many NLP datasets have been found to contain shortcuts: simple decision rules that achieve surprisingly high accuracy. However, it is difficult to discover shortcuts automatically. Prior work on automatic shortcut detection has focused on enumerating features like unigrams or bigrams, which can find only low-level shortcuts, or relied on post-hoc model interpretability methods like saliency maps, which reveal qualitative patterns without a clear statistical interpretation. In this work, we propose to use probabilistic grammars to characterize and discover shortcuts in NLP datasets. Specifically, we use a context-free grammar to model patterns in sentence classification datasets and use a synchronous context-free grammar to model datasets involving sentence pairs. The resulting grammars reveal interesting shortcut features in a number of datasets, including both simple and high-level features, and automatically identify groups of test examples on which conventional classifiers fail. Finally, we show that the features we discover can be used to generate diagnostic contrast examples and incorporated into standard robust optimization methods to improve worst-group accuracy.[1]

1 Introduction

Many NLP datasets have been found to contain shortcuts: simple decision rules that achieve surprisingly high accuracy. For example, it is possible to get high classification accuracy on paraphrase identification datasets by predicting that sentences with many common words are paraphrases of each other (Zhang et al., 2019). Such classifiers are said to be "right for the wrong reason" (McCoy et al., 2019). Shortcuts are a problem if they do not generalize to the intended test distribution (Geirhos et al., 2020). For example, paraphrase identification models might misclassify non-paraphrases that have many overlapping words.

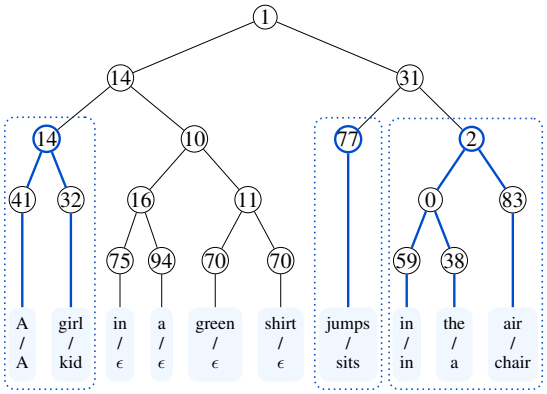
Shortcuts have been reported in many established datasets (e.g., McCoy et al., 2019; Gururangan et al., 2018; Niven and Kao, 2019; Schuster et al., 2019), as a consequence of *annotation artifacts* or so-called *spurious correlations*. Typically, these discoveries are the result of human intuition about possible patterns in a particular dataset. Our goal in this paper is to discover shortcuts automatically. If we can identify shortcuts, we can identify categories of examples on which conventional classifiers will fail, and try to mitigate these weaknesses by collecting more training data or using robust optimization algorithms.

The main challenge to automatically identifying shortcuts is to develop a formal framework for describing patterns in language data that can capture both simple and high-level features and that makes it possible to search for these patterns efficiently. Prior work has addressed only simple features like unigrams and bigrams (Wang and Culotta, 2020; Wang et al., 2022; Gardner et al., 2021), and it is difficult to extend this approach to more sophisticated patterns, like lexical overlap, without knowing the pattern in advance. Other model-based approaches use black-box interpretability methods, by using gradient-based techniques to identify tokens or training instances that influence a particular prediction (Han et al., 2020; Pezeshkpour et al., 2022; Bastings et al., 2021). These methods offer local, qualitative hints about the decision of a classifier on a particular test instance, but do not identify dataset-level features nor provide a way of measuring the strength of correlation.

Our approach is to use grammar induction to characterize and discover shortcut features (§2). Probabilistic grammars provide a principled framework for describing patterns in natural language, allowing us to formally model both simple fea-

[1]Our code for inducing grammars and finding dataset shortcuts is available at https://github.com/princeton-nlp/ShortcutGrammar.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4345 - 4363



14	77	2
A girl/A kid	jumps/sits	in the air/in a chair
Top subtrees for *entailment*		
A man/A person A man/A human	walking/walking walk/walking	in the grass/outside down the street/outside
Top subtrees for *contradiction*		
A man/A woman A woman/A man	ϵ/sitting stand/sit	at night/during the day in the air/down
Top subtrees for *neutral*		
A man/A tall human A man/An old man	ϵ/competes running/running	down the street/home in the sand/on the beach

Figure 1: Left: The most likely parse tree for an example from the SNLI validation set according to our synchronous grammar. The numbered nodes index non-terminal symbols and ϵ denotes an empty symbol. We highlight the subtrees that provide the strongest evidence in favor of the label *contradiction*, and show alternative spans generated by these non-terminals after conditioning on the class labels (Right).

tures and more sophisticated patterns. They admit tractable search algorithms and provide a natural way to measure statistics about the correlation between text patterns and labels. Grammars also offer a mechanism for identifying contrastive features, templates that appear in similar contexts across classes, but take on different values, which can be used to construct diagnostic test examples (see examples in Figure 1).

In this work, we focus on both single-sentence (e.g., sentiment analysis) and sentence-pair classification datasets (e.g., NLI, paraphrase detection). While we can use context-free grammars to model features in sentences, sentence-pair datasets present a particular challenge, as it is difficult to enumerate interactions between the pair of sentences. We propose to use synchronous context-free grammars, which formally model insertion, deletion, and alignment. We find that we can extract meaningful, dataset-specific structures that describe latent patterns characterizing these classification tasks.

We apply our approach to four classification datasets: IMDb, SUBJ, SNLI and QQP. After illustrating the shortcut features (§3), we explore whether state-of-the-art classifiers exploit these shortcuts by identifying minority examples in the datasets and then generating contrastive examples (§4). Then we demonstrate that these features can be used in robust optimization algorithms to improve generalization (§5). Finally, we compare this approach with model-based interpretability methods and n-gram features, which do not explicitly model syntax (§6). Overall, we find that grammar induction provides a flexible and expressive representation for modeling features in NLP datasets.

2 Method

2.1 Overview

We focus on text classification datasets $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$, where $y \in \mathcal{Y}$ is a categorical label and $x \in \mathcal{X}$ is either a sentence or a pair of sentences, each consisting of a sequence of words from a discrete vocabulary $\mathcal{V}$. When x is a sentence pair, we will write $x = (x^a, x^b)$ and refer to x^a and x^b as the source and target sentences, respectively.

We aim to automatically extract a description of the features that characterize the relationship between x and y, and our key idea is to define the features in terms of a *dataset-specific grammar*. Compared to a grammar extracted from a standard treebank, the grammars we induce serve as interpretable models for the distribution of sentences in the dataset. Our approach consists of two steps:

1. **Grammar induction:** First, we induce a grammar for (unlabeled) training instances $x_1, \ldots, x_N$ and get the maximum likelihood trees $t_1, \ldots, t_N$.
2. **Finding features:** We define features in terms of subtrees in the grammar, which describe patterns in the input sentences, and we search for features that have high mutual information with the class labels.

We induce one shared grammar for all classes rather than incorporating labels during grammar induction, so that the non-terminal symbols have a con-

sistent meaning across classes. This facilitates finding contrastive features. For example, in Figure 1 (right), the nonterminal symbol (77) always generates pairs of verbs, but the distribution changes according to the class label: (77) is more likely to generate walk/walking when the class label is *entailment* and more likely to generate stand/sits when the class label is *contradiction*.

2.2 Grammar Induction

In this section, we describe the two grammars we use and the training procedure.

Context-free grammar A context-free grammar (CFG) consists of an inventory of terminal symbols $\mathcal{V}$ (words), non-terminal symbols $\mathcal{N}$, and production rules of the form $\alpha \to \beta \in \mathcal{R}$, where $\alpha \in \mathcal{N}$ and $\beta \in (\mathcal{V} \cup \mathcal{N})^*$. A probabilistic CFG (PCFG) defines a distribution over trees by assigning every non-terminal symbol a categorical distribution over production rules, with the probability of a tree defined as the product of the probability of the production rules used to generate it. A PCFG defines a distribution over sequences of words $x \in \mathcal{V}^*$ by marginalizing over all trees which generate the sentence x (denoted by $\text{yield}(t)$):

$$p(x) = \sum_{t:\text{yield}(t)=x} p(t).$$

Synchronous grammar Many NLP shortcuts have been found in datasets involving pairs of sentences $x = (x^a, x^b)$. We model patterns in these datasets using a Synchronous PCFG (SCFG; Lewis and Stearns, 1968; Wu, 1997), a grammar for defining probability distributions over pairs of sequences. An SCFG assumes that both sequences were generated from a single context-free parse tree, whose terminal symbols have the form w^a/w^b, where w^a and w^b are either words in x^a or x^b respectively, or an empty symbol, denoted by ϵ, which represents a null alignment (Figure 1). SCFG productions can also be thought of as translation rules: the emission w^a/w^b represents a substitution—replacing word w^a with w^b—and null alignments represent insertion or deletion. An SCFG makes strong simplifying assumptions about the possible relationships between sentences, but, as we will show, it is still capable of modeling interesting, hierarchical structure in real-world datasets.

Parameterization and training The parameters of grammar consist of a vector θ with one entry $\theta_{\alpha\to\beta}$ for every rule $\alpha \to \beta \in \mathcal{R}$. Following Kim et al. (2019), we parameterize the grammars using a neural network. We use the neural CFG parameterization from Kim et al. (2019) and develop a similar parameterization for our SCFG, with extensions for the terminal production rules. We defer the full details to the appendix (Section A).

Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ and a grammar $\mathcal{G} = (\mathcal{V}, \mathcal{N}, \mathcal{R})$, we find a maximum likelihood estimate θ^* by maximizing the marginal likelihood of the (unlabeled) training sentences:

$$\theta^* = \arg\max_\theta \sum_{i=1}^N \log p(x_i \mid \mathcal{G}, \theta).$$

We optimize the parameters using gradient descent. After training, we calculate the maximum likelihood trees $t_1, \ldots, t_N$ for the training data, and use these trees as the basis of further analysis.

Complexity Training and parsing require enumerating the trees consistent with the input, which is calculated with the inside algorithm for CFGs (Lari and Young, 1990) and the bitext inside algorithm for SCFGs (Wu, 1997). The inside algorithm has space and time complexity of $O(|x|^3|\mathcal{G}|)$ and the bitext inside algorithm has space and time complexity of $O(|x^a|^3|x^b|^3|\mathcal{G}|)$, where $|\mathcal{G}|$ is a grammar constant determined by the number of rules in the grammar. We use a vectorized GPU implementation of the inside algorithm provided by Torch-Struct (Rush, 2020) and we implement a vectorized version of the bitext inside algorithm. The cost of the bitext parsing algorithm imposes a practical limitation on the length of the sentences we consider (Section 3.1). We also discuss possible efficient approximations in the Limitations section, which we leave as an avenue for future work.

2.3 Finding Features

The tree-annotated corpus provides a structured representation of the dataset that we can now query to find discriminative patterns at various levels of abstraction. We follow a simple procedure for finding dataset-level patterns using our tree annotations.

First, given a set of trees $t_1, \ldots, t_N$ with class labels $y_1, \ldots, y_N$, we extract the set of complete subtrees, which are subtrees whose leaves are all terminal symbols. There is at most one unique subtree for each non-terminal node in each tree, so the number of complete subtrees is roughly on the order of the number of words in the training data.

Next, we calculate the *mutual information* between each subtree and the class labels. We treat each subtree s as a boolean-valued feature function on trees, $\phi_s(t) = \mathbb{1}[s \in t]$. Let Z_s be a random variable denoting the output of ϕ_s and let Y be a random variable over $\mathcal{Y}$. The mutual information is defined as:

$$I(Z_s; Y) = \sum_{z_s \in \{0,1\}} \sum_{y \in \mathcal{Y}} p(y, z_s) \log \frac{p(y, z_s)}{p(y)p(z_s)}.$$

We estimate the mutual information between Z_s and Y using $\hat{p}(y, z_s) \propto 1 + \sum_{i=1}^{N} \mathbb{1}[y_i = y \wedge \phi_s(t_i) = z_s]$. Mutual information measures the expected amount of information we learn about Y by learning the value of ϕ_s. While we use mutual information in this paper, we could also score the features using other feature-importance metrics, such as z-score (Gardner et al., 2021).

To visualize the most discriminative patterns, we group the highest-ranked subtrees according to their root label and majority class label. Let $S(\alpha, y)$ denote the set of subtrees with root label α and majority class label y. We define a composite feature, $Z_{\alpha,y} = \bigvee_{s \in S(\alpha,y)} Z_s$ as the union of features in $S(\alpha, y)$. The result of this procedure is a concise list of class-conditional non-terminal features, which we can inspect to identify the patterns that are broadly discriminative across the dataset.

3 Finding Shortcuts

3.1 Experimental Setup

Datasets We apply our approach to two single-sentence and two sentence-pair classification datasets. **IMDb** (Maas et al., 2011) is a binary sentiment analysis dataset consisting of paragraph-length movie reviews. **SUBJ** (Pang and Lee, 2004) is a subjectivity classification dataset, containing sentences labeled as either *subjective* or *objective*. **SNLI** (Bowman et al., 2015) is a three-way classification task; given two sentences, x^a and x^b, the objective is to determine whether x^a *entails* x^b, *contradicts* x^b, or is *neutral*. **QQP** (Iyer et al., 2017) consists of pairs of questions from quora.com labeled as being either *paraphrases* or *not paraphrases*.

For all experiments, we fix the size of the grammar to be 96 non-terminals symbols, divided into 32 internal non-terminal symbols and 64 pre-terminal symbols, similar to Kim et al. (2019), and use a lowercase word-level tokenizer with a maximum vocabulary size of 20,000 words. For the sentence-pair datasets, we randomly sample 65K/16K class-balanced sets of training/validation examples that fit within the length limit imposed by the bitext inside-outside algorithm ($|x^a| \times |x^b| < 225$). This length limit covers approximately 80% of SNLI and 70% of QQP. For IMDb, we split the movie reviews into sentences for the purposes of training the PCFG, but compute feature statistics using the full reviews. More implementation details are in Appendix B.

3.2 A Look at the Top Features

In this section, we qualitatively explore some of the dataset-level shortcuts we find. Our procedure for each dataset is the same: given a set of trees, we enumerate all complete subtrees and sort them by mutual information, treating each subtree as a binary indicator variable. Then we group the subtree features by root label and majority class label and inspect the most informative groups of subtrees. The majority class label for a feature Z is defined as the most common class label among training examples for which $Z = 1$. For each feature Z, we report the number of training examples for which $Z = 1$ (**Count**) and the percentage of these having the majority class label (**% Majority**). We present the most interesting results here and include extended results in Appendix D.1.

IMDb Not surprisingly, we find that the most informative features in IMDb include adjectives, adverbs, and nouns with high sentiment valence. We highlight some of the more interesting patterns in Table 1. For example, we discover that negative reviews are almost three times as likely as positive reviews to mention the length of the film (node ⑧), and we find that the grammar has learned a clear category corresponding to names (node ⑤).

We also confirm that the grammar recovers a known shortcut in IMDb, numerical ratings (Ross et al., 2021; Pezeshkpour et al., 2022). We find that a single non-terminal (node ㉙) is responsible for generating ratings, including both simple, numerical ratings, which have been documented in earlier work, as well as letter grades and ratings on a star system (Table 2).

SUBJ In the SUBJ dataset (Table 3), the most informative features reflect how this dataset was constructed (Pang and Lee, 2004): the *subjective* class consists of movie reviews from Rotten Tomatoes and the *objective* class consists of movie sum-

	Root	Description	Patterns	Count	% Majority
N	⑤	Negative actors	ed wood , steven seagal , uwe boll , van damme , tom savini	174	95.5
	㉙	Negative ratings	4 / 10 , 3 / 10 , 1 / 10 , 2 / 10 , 1 / 2 from * * * *	429	96.8
	⑧	Negative durations	30 minutes , 10 minutes , five minutes , 90 minutes , 2 hours	1,412	76.7
P	⑤	Positive actors	walter matthau , jon voight , james stewart , william powell , philo vance	751	88.6
	㉙	Positive ratings	10 / 10 , 8 / 10 , 7 / 10 , highly recommended . , 9 / 10	486	98.8
	⑧	Positive durations	many years	95	69.1

Table 1: Six high-scoring features in IMDb, grouped by majority class (N: Negative, P: Positive) and showing at most five spans per row, with our own descriptions of the pattern reflected in each row.

	Patterns	Count	% Majority
N	3 out of 10 . , 4 out of 10 . , 1 out of 10 .	41	100.0
	my grade : d , my grade : f , my grade : c	33	100.0
	1 / 2 from * * * *	44	93.2
P	10 out of 10 . , 7 out of 10 . , 8 out of 10.	51	100.0
	my vote is eight . , my vote is seven .	40	100.0

Table 2: Additional realizations of the "movie rating" pattern in IMDb (N: negative, P: positive). All of these spans correspond to subtrees for root ㉙.

	Root	Patterns	Count	% Majority
S	㉗	a movie , the film , the movie , this movie	980	86.3
	③	comes off , ' s hard , makes up , ' d expect	460	87.4
O	㉗	his life , his wife , his father , his mother	1,628	80.1
	③	finds himself , finds out , falls in love , is [UNK]	205	85.5

Table 3: Four high-scoring features in SUBJ, filtering to subtrees with depth of at least 2, grouped by majority class (S: subjective, O: objective).

maries from IMDb. A similar observation was made by Zhong et al. (2022), who trained a neural network to generate natural language descriptions of the differences between text distributions. Our method points us to the same conclusion, but by modeling the statistics of the dataset rather than querying a black-box neural network.

SNLI Now we consider the synchronous grammar features for SNLI (Table 4). Prior work has documented the presence of hypothesis-only shortcuts in SNLI as well as shallow cross-sentence features like lexical overlap (McCoy et al., 2019; Gururangan et al., 2018; Poliak et al., 2018). The SCFG features reveal a number of more sophisticated patterns and clusters them in clear categories. The *contradiction* class has the most highly discriminative shortcut features, which mainly involve replacing a subject or verb word with a direct **antonym**. The most informative *neutral* features involve **additions**, such as adding adjectives or prepositional phrases. The highest scoring *entailment* examples include **hypernyms**, such as changing "man" to "human". These features explicitly model the alignment between grammatical roles, as well as insertion and deletion, giving a high-level view of some of the common strategies employed by crowdworkers in creating this dataset.

Additional, higher-level features are presented in Appendix Table 15, which lists the internal production rules with the highest mutual information. In general, these features are less discriminative than lexicalized features, but they describe more abstract properties, such as removing prepositional phrases from x^a to create an entailment.

QQP The highest scoring features in QQP are listed in Table 5. The best known shortcut in QQP is lexical overlap, which is more likely to be high between paraphrases, and the SCFG features echo this fact: most of the highest ranking *paraphrase* features are pairs of aligned words or phrases, and the highest ranking *no paraphrase* features are function words that have no alignment in the corresponding question. Other prominent features involve changes to the question structure, as well as a number of specific topics that are surprisingly prevalent in the dataset and provide strong evidence that a pair of questions are paraphrases, including open-ended discussion topics such as New Year's resolutions, World War 3, and the 2016 presidential election, as well as lifestyle advice about how to make money or lose weight.

4 Do Models Exploit These Shortcuts?

In this section, we explore how the shortcuts we have discovered affect the generalization behavior of conventional classifiers that are trained on the same data. We train BERT-base (Devlin et al.,

	Root	Description	Patterns	Count	% Majority
E	(44)	Copy verb	walking/walking , running/running , playing/playing , sitting/sitting , jumping/jumping	1,326	68.4
	(14)	Subject phrase hypernym	a man/a person , a man/a man , a woman/a person , man/a man , a man/a human	9,009	45.7
	(4)	Expletive construction	a /there is , ϵ/there are , two /there are , a /there are , ϵ/there is	1,725	63.0
C	(32)	Subject antonym	man/woman , woman/man , boy/girl , dog/cat , girl/boy	1,235	91.1
	(14)	Subject phrase antonym	a man/a woman , a woman/a man , a man/ nobody , a boy/a girl , a dog/a cat	1,351	82.5
	(78)	Verb antonym	standing/sitting , walking/sitting , sitting/standing , walking/running , running/sitting	695	92.6
	(41)	Definite article	a/the , ϵ/the	15,436	39.2
	(85)	Adjective antonym	black/white , red/blue , ϵ/empty , ϵ/living , white/black	560	76.9
N	(49)	Added function word	ϵ/to , ϵ/for , ϵ/a , ϵ/the , ϵ/his	14,478	50.5
	(35)	Added object	ϵ/[UNK] , ϵ/work , ϵ/get , ϵ/friends , ϵ/park	4557	59.8
	(85)	Added adjective	ϵ/tall , ϵ/sad , ϵ/[UNK] , ϵ/new , ϵ/big	1,945	72.1
	(17)	Added PP phrase	ϵ/to work , ϵ/to get , ϵ/to buy , ϵ/the park , ϵ/on vacation	1,411	71.4

Table 4: Twelve of the highest scoring features in SNLI, grouped by majority class (E: Entailment, C: Contradiction, N: Neutral) with our own descriptions of the pattern reflected in each row. ϵ stands for the empty string. For each feature, we report the number of training examples and the percentage having the majority class label.

	Root	Description	Patterns	Count	% Majority
N	(70)	Additions	ϵ/[UNK] , ϵ/in , ϵ/a , ϵ/- , ϵ/for	23,987	60.7
	(49)	Deletions	[UNK]/ϵ , in/ϵ , a/ϵ , like/ϵ , of/ϵ	21,299	61.6
	(59)	Change question word	why/how , why/what , how/why , why/can , what/is	3,348	70.8
P	(14)	How-to questions	how can/how can , how do/how can , how can/how do , how do/how do , how can /what is the	9,684	66.2
	(25)	Discussion topics	new year/new year , world war/world war , donald trump/donald trump , hillary clinton/hillary clinton , long distance/long distance	2,179	82.9
	(59)	Same question word	how/how , why/why , when/when , how/what	17,264	60.3

Table 5: Six of the highest scoring features in QQP, grouped by majority class (N: non-paraphrase, P: paraphrase) with our own descriptions of the pattern reflected in each row. ϵ stands for the empty string. For each feature, we report the number of training examples and the percentage having the majority class label.

2019) and RoBERTa-base (Liu et al., 2019) classifiers on our SNLI and QQP training splits and examine the performance by finding minority groups of counter-examples in the dataset, and then by designing contrastive examples using the grammar.

Observational counter-examples First, for each shortcut feature Z, let $y^{Z=1}$ denote the majority label for training instances that contain the shortcut feature ($Z = 1$). We take the validation examples x_i, y_i for which $Z = 1$ and partition them into *supporting examples* and *counter-examples* according to whether or not $y_i = y^{Z=1}$. We report the accuracy of the BERT and RoBERTa models on supporting and counter-examples in Table 6. Both models consistently perform higher than average on the supporting examples and much worse on the counter-examples, indicating that the grammar feature are at least correlated with the features used by these classifier. This trend is consistent for all the features, perhaps suggesting that these models exploit every available feature to some extent.

Creating contrastive examples In the previous section, we established that there is a correlation between shortcut features and BERT error rates using counter-examples that appear in the training data. In this section, we generate controlled contrasting examples to test specific hypotheses about what function the model has learned. Our procedure in this section is similar to counter-factual data augmentation (Kaushik et al., 2020) and contrast sets (Gardner et al., 2020). The difference is that our edits are based on explicit feature representations derived from the dataset, allowing us to better control for confounding features.

We focus on the three patterns we highlighted in SNLI: simple hypernyms, antonyms, and additions. For each feature Z with majority label $y^{Z=1}$, we design a rule-based edit that will select and perturb existing validation instances (x, y) to obtain instances (x^*, y^*) for which $Z = 1$ but $y^* \neq y^{Z=1}$. **Hypernyms:** We select *neutral* or *contradiction* examples that have an aligned subject word (e.g. man/man), replace the hypothesis subject word with a hypernym, and expect the label to stay the same. **Antonyms:** Pick *entailment* or *neutral* examples with an adjective modifying the subject noun, add an antonym adjective to an object noun, and expect the label to become *neutral*. **Add adjective:** Pick *contradiction* examples, add an adjective

		BERT		RoBERTa	
		S	C	S	C
SNLI	**Entailment**				
	Copy verb	98.6	83.6	98.6	83.6
	Subj. phrase hypernym	92.5	80.3	92.4	83.8
	Expletive construction	97.2	77.8	96.4	79.1
	Contradiction				
	Subj. antonym	96.9	61.5	97.3	76.9
	Subj. phrase antonym	98.2	77.8	98.2	82.5
	Verb antonym	99.4	55.6	98.8	66.7
	Definite article	86.2	82.2	88.9	82.9
	Adjective antonym	91.7	71.4	95.8	78.6
	Neutral				
	Added function word	86.6	81.0	89.3	81.6
	Added object	89.0	76.1	93.0	76.1
	Added adjective	94.4	69.2	94.2	74.1
	Added PP phrase	92.9	77.0	95.0	77.0
QQP	**Non-paraphrase**				
	Deletion	85.4	86.8	86.9	86.9
	Addition	86.8	85.1	87.8	85.5
	Change question word	87.4	81.2	89.6	78.8
	Paraphrase				
	How-to questions	90.3	71.2	92.3	73.0
	Discussion topics	98.4	56.9	96.2	66.7
	Same question word	90.3	75.0	91.1	76.9

Table 6: We find the validation examples in SNLI and QQP containing each shortcut and partition them into **Supporting** examples (S) and **Counter**-examples (C) according to whether or not they have the training majority class label, and report the accuracy of BERT and RoBERTa models on each split. (See Section 4.)

modifying the subject noun, and expect the label to stay the same. In each case, we use the grammar to identify the set of possible edits. That is, we select antonyms like white/black or small/large and add adjectives like ϵ/tall and ϵ/sad .

For each validation example (x_i, y_i), we create a contrast set $\mathcal{S}_i$ consisting of one or more perturbations of x_i. For example, if x_i contained man/man , $\mathcal{S}_i$ will include examples containing man/person and man/human . We report the **error rate**, defined as the percentage of the contrast sets $\mathcal{S}_i$ for which the model predicts $y^{Z=1}$ for any $x^* \in \mathcal{S}_i$, restricted to sets such that the model classified the original (x_i, y_i) correctly.

The results of this experiment are in Table 7, along with an example of each edit. On each test set, we find many perturbations that lead the model to change its prediction. The model performs worst on the Antonyms test set, suggesting that the presence of contradicting adjectives may be a strong signal to the model, regardless of whether the adjectives are attached to the same entity.

Edit	# Sets	Error
Hypernyms A man is smoking at sunset. A ~~man~~ +person smoking a cigarette.	389	$21.8_{\pm 0.8}$
Antonyms Two black dogs splash around on the beach. The dogs are playing with a +white ball.	281	$71.1_{\pm 3.8}$
Add adjective A man taking photos of nature. A +sad man is taking photos of a wedding.	1,470	$45.6_{\pm 8.4}$

Table 7: Examples of the contrastive edits we create for three shortcut features, the number of contrast sets (each consisting of one or more perturbations of a single validation instance), and the **error rate** (Section 4). We report the average and standard deviation of BERT models trained with three random seeds.

Method	Hypernyms	Antonyms	Add adj.
BERT	$21.8_{\pm 0.8}$	$71.1_{\pm 3.8}$	$45.6_{\pm 8.4}$
JTT	$21.5_{\pm 0.8}$	$69.7_{\pm 3.4}$	$39.3_{\pm 7.8}$
DRiFt	$\mathbf{13.0}_{\pm 4.1}$	$\mathbf{68.7}_{\pm 6.8}$	$\mathbf{29.4}_{\pm 4.5}$

Table 8: Evaluating robust training methods on the test sets described in Table 7. We report the mean error rate (↓) and standard deviation from three runs (Section 5).

5 Remedying Shortcuts

Once we have found shortcuts, to what extent can we can mitigate them using standard robust optimization algorithms? We conduct a small-scale experiment, focusing the contrasting examples we created for SNLI. We compare two methods: Just Train Twice (JTT; Liu et al., 2021) and DRiFt (He et al., 2019). Note that JTT does not assume known shortcut features while DriFT does. Specifically, JTT upsamples the subset of training examples that are misclassified by a weak model (we use BERT-base trained for one epoch). DRiFT takes as input a biased model and trains a new model with a regularization term that effectively upweighs the reward for instances that the biased model misclassifies (we set the biased model to be a logistic regression classifier trained on feature vectors indicating the set of SCFG production rules that appear in each tree). Full details are provided in Appendix B.

Results Table 8 shows that the robust tuning methods improve performance on all the test sets, with DRiFt achieving a lower error rate than JTT. However, the improvements on the Antonyms test set are less substantial, which could be explained by the fact that this shortcut has few counter-examples in the training data. These results suggest that the

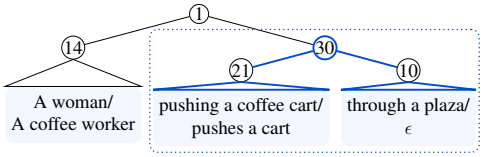
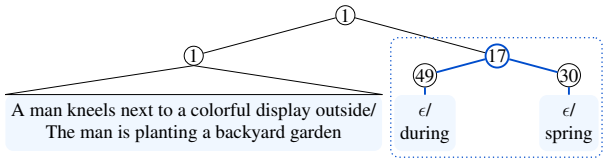
Saliency map	Tree parse	Label	Prediction
A woman pushing a coffee cart through a plaza. A coffee worker pushes a cart.		Neutral	Entailment
A man kneels next to a colorful display outside. The man is planting a backyard garden during spring .		Contradiction	Neutral

Table 9: Saliency maps and SCFG parse trees for two SNLI validation instances that are misclassified by a fine-tuned BERT model (Section 6). For each tree, we highlight the production rule most strongly correlated with the predicted label, which is associated with the pattern of either deleting or inserting a prepositional phrase. In both cases, when we delete the prepositional phrase, the model predicts the correct label.

best solution to addressing highly correlated shortcuts may be to collect additional training data.

6 Comparing Existing Methods for Finding Shortcuts

Saliency methods In Table 9, we compare our grammar-based approach with saliency heatmaps, as a method for identifying the discriminative features in an individual example. We show two validation examples that a BERT-based classifier fails to classify. Following Bastings et al. (2021), we use the L2 norm of the Integrated Gradient score (Sundararajan et al., 2017) and highlight the five tokens with the highest score. The heatmap highlights input words but does not provide information about how different input words are connected.

To find locally important features using the grammar, we find the maximum likelihood parse trees and list the production rules in order of $\hat{p}(\hat{y} \mid r)/\hat{p}(y^* \mid r)$, where y^* is the true label, $\hat{y}$ is the predicted label, and $\hat{p}(y \mid r)$ is proportional to the number of training trees with label y that contain production rule r. The most discriminative rules include production rules associated with deleting a prepositional phrase (which appear twice as often in *entailment* examples) and inserting a prepositional phrase (more common in *neutral* examples). In both cases, when we delete the prepositional phrase, the model predicts the correct label.

N-gram-based methods The grammar features can capture information that cannot be expressed with n-gram features, such as alignment and syntactic roles, but how relevant is this information for diagnosing dataset shortcuts? We explore this question in Figure 2 by comparing the SNLI features described in Table 4 to simpler n-gram features created by discarding information about syntax and alignment. For example, the *Subject hypernym* feature appears in sentence pairs like "A man is walking/A human is walking." We compare this feature to the corresponding n-gram pair feature, which has a value of true whenever "man" appears in the premise and "human" appears in the hypothesis, and would include sentences like "A man is feeding ducks/A man is feeding a human." Finally, prior work has reported that individual premise words are correlated with labels in NLI datasets (Poliak et al., 2018; Gururangan et al., 2018), so we also compare these features with the hypothesis-only feature, which is true whenever "human" appears in the second sentence. Empty alignments, e.g. ϵ/w^b, are less straightforward to compare; we define the equivalent n-gram pair feature as $w^b \in x^b$.

Figure 2 shows that, as we consider increasingly simple features, we identify more examples that contain the shortcut, but the shortcut becomes less discriminative, and has a weaker correlation with the BERT classifier's accuracy: BERT performs relatively worse on the supporting examples, and better on the counter-examples, indicating that these features may be less useful for diagnosing classifier errors. More details are in Appendix C.

7 Related Work

Finding shortcuts Prior work has identified shortcuts in NLP datasets by developing diagnostic evaluation datasets (McCoy et al., 2019; Niven and Kao, 2019; Rosenman et al., 2020), training partial input baselines (Gururangan et al., 2018; Poliak et al., 2018), or calculating statistics of simple features, like n-grams, that can be enumerated explic-

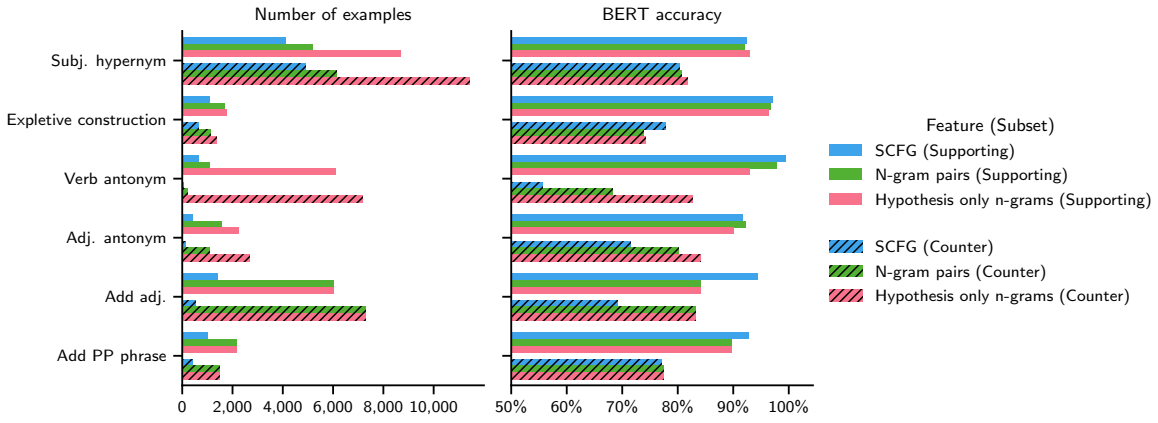


Figure 2: We compare the SCFG features from SNLI with equivalent n-gram feature that discard the alignment information provided by the grammar (Section 6). As we consider simpler features, the features appear in more examples but become less discriminative, and tend to have a weaker correlation with BERT accuracy.

itly (e.g. Schuster et al., 2019; Gardner et al., 2021). Han et al. (2020) use influence functions (Koh and Liang, 2017) to retrieve training instances that are relevant to a model's prediction. Pezeshkpour et al. (2022) combine instance attribution with feature attribution, which highlights salient input tokens. In contrast, our aim is to find dataset-level shortcuts that can be expressed as explicit feature functions.

Defining spurious correlations A related line of work has addressed the question of which correlations should be considered spurious. Gardner et al. (2021) suggest that all correlations between labels and low-level features, such as unigrams, are spurious. Eisenstein (2022) argues that such correlations will arise naturally in most language classification settings, and domain expertise is needed to determine which correlations might be harmful. Our aim is to provide a summary of the shortcuts in a dataset, so that a practitioner can determine which are undesirable and remedy them if needed.

Feature importance Our approach is related to methods for identifying important input features, like LIME (Ribeiro et al., 2016) . Lundberg and Lee (2017) present a framework for measuring local feature importance based on Shapley values (Shapley, 1953), which can be applied to both model-specific and model-agnostic notions of importance, and Covert et al. (2020) extend this approach to global importance scores. These methods require first identifying the set of features, while our focus is on inducing a richer set of features. We use a simple metric for feature importance, and leave more sophisticated metrics to future work.

Grammar induction Grammar induction has been a long-standing subject of research in artificial intelligence. Prior work has used grammar induction for linguistic analysis (Johnson, 2008; Dunn, 2018) and feature extraction (Hardisty et al., 2010; Wong et al., 2012). Synchronous and quasi-synchronous grammars (Smith and Eisner, 2006) have been used in machine translation and applied to a variety of other NLP tasks (e.g. Wang et al., 2007; Blunsom et al., 2008; Yamangil and Shieber, 2010), but have largely been supplanted by end-to-end neural network approaches. Kim (2021) develop a neural quasi-synchronous grammar as an interpretable, rule-based model for sequence transduction tasks. Our work shares a similar motivation, but applied to a different goal, modeling dataset shortcuts.

8 Conclusion

We have developed an approach for automatically finding dataset shortcuts by inducing dataset-specific grammars. We demonstrated that it reveals interesting shortcut features in four classification datasets and can be used as a diagnostic tool to identify categories of examples on which classifiers are more likely to fail. Future work will explore approximate inference methods to scale this approach to datasets with longer sequences, and extensions to more expressive grammar formalisms.

Acknowledgements

We thank the members of the Princeton NLP group and the anonymous reviewers for their valuable comments and feedback.

Limitations

Our method has several limitations that raise interesting challenges for future work.

Grammar scalability First is the scalability of the synchronous parsing algorithm, which limits us in practice from applying this approach to datasets with long sentence pairs. One direction for future work is to explore approximate inference techniques that will allow these methods to scale to large datasets with longer sequences and use fewer computational resources.

Grammar expressiveness Second is the expressiveness of the grammar, which assumes that pairs of sentences can be modeled with a single parse tree. This assumption works well for relatively simple datasets like SNLI, but is less reasonable for sentences that have very different syntactic structure. An interesting direction for future work is to explore more expressive grammar formalisms, such as synchronous tree substitution grammars (Shieber and Schabes, 1990).

Comparing shortcut finding methods There are many existing approaches to finding shortcuts as we discussed in the paper, but it is difficult to have an apples-to-apples comparison. In particular, attribution-based methods (Han et al., 2020; Pezeshkpour et al., 2022) do not provide explicit feature representation of shortcuts, and so it is difficult to say whether these methods are capable of identifying the same patterns. We have shown qualitative comparisons of the kinds of features provided by different methods but leave the question of better quantitative evaluation to future work.

Applications to other languages and tasks We only apply our approach to four English-language datasets. In particular, the sentence pair datasets we consider are based on sentence similarity judgments, where synchronous grammars are a good choice. We hope to apply the approach to other tasks and other languages in the future and explore formalisms that are better suited to modeling more diverse relationships between strings.

Out-of-domain generalization We conducted one set of experiments with robust optimization and showed some improvements on in-domain minority examples. In future work, we are also interested in exploring whether our shortcut-finding methods can be useful for improving performance in generalization to other out-of-domain distributions.

References

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2021. "Will you find these shortcuts?" A protocol for evaluating the faithfulness of input salience methods for text classification. *arXiv preprint arXiv:2111.07367*.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. Bayesian synchronous grammar induction. *Advances in Neural Information Processing Systems (NeurIPS)*, 21.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642.

Ian Covert, Scott M Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:17212–17223.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Jonathan Dunn. 2018. Finding variants for construction-based dialectometry: A corpus-based approach to regional CxGs. *Cognitive Linguistics*, 29(2):275–311.

Jacob Eisenstein. 2022. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4326–4331.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1307–1323.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E Peters, Alexis Ross, Sameer Singh, and Noah A Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1801–1813.

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.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 107–112.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Association for Computational Linguistics (ACL)*, pages 5553–5563.

Eric Hardisty, Jordan Boyd-Graber, and Philip Resnik. 2010. Modeling perspective using adaptor grammars. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 284–292.

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.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First Quora dataset release: Question pairs. *quora.com*.

Mark Johnson. 2008. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations (ICLR)*.

Yoon Kim. 2021. Sequence-to-sequence learning with latent neural grammars. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:26302–26317.

Yoon Kim, Chris Dyer, and Alexander M Rush. 2019. Compound probabilistic context-free grammars for grammar induction. In *Association for Computational Linguistics (ACL)*, pages 2369–2385.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pages 1885–1894. PMLR.

Karim Lari and Steve J Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech & Language*, 4(1):35–56.

Philip M Lewis and Richard Edwin Stearns. 1968. Syntax-directed transduction. *Journal of the ACM (JACM)*, 15(3):465–488.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just Train Twice: Improving group robustness without training group information. In *International Conference on Machine Learning (ICML)*, pages 6781–6792. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 142–150.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, pages 3428–3448.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Association for Computational Linguistics (ACL)*, pages 4658–4664.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Association for Computational Linguistics (ACL)*, pages 271–es.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research (JMLR)*, 12:2825–2830.

Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron C Wallace. 2022. Combining feature and instance attribution to detect artifacts. In *Findings of Association for Computational Linguistics (ACL)*, pages 1934–1946.

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

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*, pages 1135–1144.

Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. 2020. Exposing shallow heuristics of relation extraction models with challenge data. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 3702–3710.

Alexis Ross, Ana Marasović, and Matthew E Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3840–3852.

Alexander M Rush. 2020. Torch-Struct: Deep structured prediction library. In *Association for Computational Linguistics (ACL)*, pages 335–342.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 28:307–317.

Stuart M Shieber and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *International Conference on Computational Linguistics (COLING)*.

David A Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 23–30.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pages 3319–3328. PMLR.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32.

Tianlu Wang, Diyi Yang, and Xuezhi Wang. 2022. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1719–1729.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 3431–3440.

Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. 2012. Exploring adaptor grammars for native language identification. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 699–709.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Elif Yamangil and Stuart M Shieber. 2010. Bayesian synchronous tree-substitution grammar induction and its application to sentence compression. In *Association for Computational Linguistics (ACL)*, pages 937–947.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1298–1308.

Ruiqi Zhong, Charlie Snell, Dan Klein, and Jacob Steinhardt. 2022. Summarizing differences between text distributions with natural language. In *International Conference on Machine Learning (ICML)*.

A Grammar Parameterization

Let $\mathcal{N}, \mathcal{V}, \mathcal{R}$ denote the set of non-terminal symbols, terminal symbols, and production rules in the grammar. We only consider grammars in binary normal form. Following Kim et al. (2019), we designate a subset $\mathcal{P} \subseteq \mathcal{N}$ as pre-terminals. R is then defined as all rules of the form $\alpha \rightarrow w$, where $\alpha \in \mathcal{P}$ and $w \in \mathcal{V}$, or $\alpha \rightarrow \beta\ \gamma$, where $\alpha \in \mathcal{N} \setminus \mathcal{P}$ and $\beta, \gamma \in \mathcal{N}$, as well as $s \rightarrow \alpha$, where $\alpha \in \mathcal{N}$ and S is the start symbol.

We parameterize our grammars using neural networks, following Kim et al. (2019) and Kim (2021). For the PCFG, we use the parameterization from the neural PCFG of Kim et al. (2019):

$$\begin{aligned} p(S \rightarrow \alpha) \quad & \propto \exp(f_1(\mathbf{v}_S)^\top \mathbf{u}_{S \rightarrow \alpha}) \\ p(\alpha \rightarrow \beta\ \gamma) \quad & \propto \exp(f_2(\mathbf{v}_\alpha)^\top \mathbf{u}_{\beta\gamma}) \\ p(\alpha \rightarrow w) \quad & \propto \exp(f_3(\mathbf{v}_\alpha)^\top \mathbf{u}_w), \end{aligned}$$

where $\mathbf{v}_*, \mathbf{u}_* \in \mathbb{R}^d$ are embeddings and f_1, f_2, f_3 are multilayer perceptrons with two hidden layers and residual connections.

For the SPCFG, we parameterize the the starting rules $S \rightarrow \alpha$ and the binary productions $\alpha \rightarrow \beta\ \gamma$ the same as in the PCFG. We parameterize the terminal productions $\alpha \rightarrow w$ by factoring according to the kind of emission, $k \in \{w^a/w^b, w^a/\epsilon, \epsilon/w^b, \text{copy}\}$, where ϵ is the empty symbol and copy is a production with w^a/w^b with $w^a = w^b$. The emission distribution is then factored as:

$$\begin{aligned} p(\alpha \rightarrow w^a/w^b) = p(k \mid \alpha) \times p(w^a \mid \alpha, k) \\ \times\ p(w^b \mid \alpha, k, w^a). \end{aligned}$$

We parameterize these rules as in the PCFG:

$$\begin{aligned} p(k \mid \alpha) \quad & \propto \exp(f_3(\mathbf{v}_{\alpha^k})^\top \mathbf{u}_k) \\ p(w^a \mid k, \alpha) \quad & \propto \exp(f_4(\mathbf{v}_{\alpha^a})^\top \mathbf{u}_{w^a}) \\ p(w^b \mid k, \alpha, w^a) \quad & \propto \exp(f_5(\mathbf{v}_{\alpha^b} + \mathbf{v}_{w^a})^\top \mathbf{u}_{w^b}), \end{aligned}$$

with restrictions to ensure that the distribution is of the emission kind k.

B Training Details

Grammar induction For all of our experiments, we fix the size of the grammar at 64 pre-terminal symbols and 32 non-terminal symbols. We set the hidden dimension d of the grammar embeddings to be 256 and define every f_i to be a ReLU network with two hidden layers, following Kim et al. (2019). We use a learning rate of 1e-3 and the Adam optimizer. For the PCFGs, we train for up to 40 epochs, evaluating on validation data every 4096 steps and stopping early after five checkpoints with no improvement in validation loss (negative log likelihood). For the SCFGs, we train for up to 10 epochs and use the same early stopping policy.

We run our main experiments on four RTX 3090 Ti GPUs with 24GB of memory each, but also test our SCFG implementation on RTX 2080 GPUs with 12GB of memory. With four GPUs, and training on 64,000 examples with $|x^a| \times |x^b| \leq 225$, running SCFG grammar induction until convergence and then parsing all of the training and validation examples takes between 12 and 24 hours hours. For the PCFGs, we use a mini-batch size of four sequences per GPU, and for the SCFG we use a mini-batch size of one pair of sequences per GPU. Consistent with the report of Kim (2021), we find that GPU memory is the main bottleneck to scaling the SCFG.

Classification experiments We train BERT and RoBERTa classifiers using a learning rate of 1e-5 and do not tune any hyperparameters. The biased model for DRiFt is a logistic regression model trained with L1 regularization using the default implementation in scikit-learn (Pedregosa et al., 2011), and the identification model for Just-Train-Twice is a BERT model trained for one epoch. We use a mini-batch size of 16 and training these models for up to 20 epochs, evaluating on the validation set every 4096 steps, stopping early after five checkpoints with no improvement in validation accuracy.

C Comparing Simple Features

In Table 10, we compare our SCFG features to corresponding n-gram features in terms of prevalence, accuracy, and the correlation with BERT's error rate on SNLI and QQP. The aim of this experiment is to estimate the extent to which the BERT classifiers exploit the type of syntactic patterns identified by the grammar or, for example, discard positional information.

D Additional Analysis

D.1 Additional Features

We list additional features for IMDb (Table 11), SUBJ (Table 12), SNLI (Table 13), and QQP (Ta-

		Number of examples						*BERT accuracy*					
		SCFG		**N-gram pairs**		**Hyp. only**		**SCFG**		**N-gram pairs**		**Hyp. only**	
		S	C	S	C	S	C	S	C	S	C	S	C
SNLI	**Entailment**												
	Subj. phrase hypernym	4,115	4,894	5,204	6,149	8,698	11,427	92.5	80.3	92.0	80.8	93.0	81.8
	Copy verb	908	418	3,151	2,990	6,224	9,884	98.6	83.6	95.5	81.5	92.7	82.0
	Expletive construction	1,087	638	1,682	1,112	1,782	1,368	97.2	77.8	96.7	73.9	96.4	74.2
	Contradiction												
	Subj. antonym	1,127	108	2,178	1,121	10,243	18,129	96.9	61.5	95.5	81.1	88.2	84.2
	Subj. phrase antonym	1,116	235	1,784	1,148	9,156	15,327	98.2	77.8	94.0	83.8	88.8	84.7
	Verb antonym	645	50	1,107	215	6,104	7,188	99.4	55.6	97.8	68.2	93.1	82.6
	Definite article	6,055	9,381	10,803	17,470	10,803	17,470	86.2	82.2	80.3	85.7	80.3	85.7
	Adj. antonym	432	128	1,565	1,099	2,262	2,658	91.7	71.4	92.3	80.1	90.1	84.1
	Neutral												
	Add function word	7,317	7,161			21,024	40,092	86.6	81.0			78.9	87.7
	Add object noun	2,964	1,711			16,343	29,792	89.0	76.1			80.1	86.9
	Add adjective	1,404	541			6,044	7,298	94.4	69.2			84.1	83.1
	Add PP phrase	1,009	402			2,186	1,473	92.9	77.0			89.7	77.5
QQP	**Paraphrase**												
	Discussion topic	1,806	373	2,210	480	2,891	963	98.4	56.9	98.3	58.1	98.2	68.3
	How-to question	6,415	3,269	6,964	3,587	17,123	14,370	90.3	71.2	90.5	71.7	90.6	78.5
	Same question word	10,403	6,861	11,485	7,634	25,302	24,279	90.3	75.0	90.3	75.1	89.5	80.6

Table 10: We compare the SCFG features from SNLI (top) and QQP (bottom) with an equivalent pair-of-n-gram feature that discards the alignment information provided by the grammar (see Section 6). For example, let x^a, x^b denote the first and second sentence in a pair, and t denote the maximum likelihood SCFG tree. In the first row, the **SCFG** feature represents the indicator $\mathbb{1}[(\text{(14) a man/a person}) \in t \vee (\text{(14) a woman/a human}) \in t \vee \ldots]$; the **N-gram pair feature** represents the indicator $\mathbb{1}[(\text{a man} \in x^a \wedge \text{a person} \in x^b) \vee (\text{a woman} \in x^a \wedge \text{a human} \in x^b) \vee \ldots]$; and the Hypothesis-only n-gram feature (**Hyp. only**) represents $\mathbb{1}[(\text{a person} \in x^b) \vee (\text{a human} \in x^b) \vee \ldots]$. In the case of an empty alignment, e.g. w^a/ϵ, the equivalent n-gram feature is defined as $w^a \in x^a$. For each feature Z, we find the examples for which $Z = 1$ and partition them into supporting examples (**S**) and counter-examples (**C**) according to whether or not they have the class label y that appears most often in the training example for which $Z = 1$. We report the number of training examples in each subset and the accuracy of a BERT classifier on corresponding validation examples. The simpler features appear in more examples but tend to be less discriminative and to have a weaker correlation with the BERT classifier's accuracy: BERT performs relatively worse on the supporting examples, and better on the counter-examples, indicating that these features may be less useful for diagnosing classifier errors. In QQP, the grammar features do not convey much more information than n-gram pairs, perhaps indicating that syntactic alignment is relatively unimportant for identifying paraphrases in this dataset.

ble 14). For each table, we pick the 1,000 subtrees with the highest mutual information scores (restricted to subtrees with two or more leaves) and group them by root label α and majority class label y. For each subtree s, the corresponding feature Z_s is the boolean feature defined as 1 if subtree s appears in the maximum likelihood parse tree and 0 otherwise, and we calculate the mutual information using the empirical likelihoods, $\hat{p}(y, s) \propto 1 + \sum_{t_i, y_i} \mathbb{1}[y_i = y \wedge s \in t_i]$, where t_i represents the parse for input x_i. Each row represents a composite feature $Z_{\alpha,y} = \bigvee_{s \in S(\alpha,y)} Z_s$, where S is the subset of the top 1,000 subtrees that have root label α and majority class label y. We list the rows in decreasing order of $I(Z_{\alpha,y}; Y)$. For each row, we report the number of training examples with $Z_{\alpha,y} = 1$ for each class label, and list the spans corresponding to up to five subtrees $s \in S(\alpha, y)$, in decreasing order of $I(Z_s; Y)$.

D.2 Higher Level Features

Table 15 lists the twelve binary production rules $\alpha \rightarrow \beta\ \gamma$ that have the highest mutual information in SNLI. For each rule r, the corresponding feature Z_r is the boolean feature defined as 1 if production rule r appears in the maximum likelihood parse tree and 0 otherwise, and we calculate the mutual information using the empirical likelihoods, $\hat{p}(y, r) \propto 1 + \sum_{t_i, y_i} \mathbb{1}[y_i = y \wedge r \in t_i]$, where t_i represents the parse for input x_i. The most informative features include removing a prepositional phrase (entailment) and adding an object or prepositional phrase (neutral).

D.3 Contrast Sets

In addition to creating rule-based contrast sets, we also create a set of contrasting examples manually according to the following procedure, illustrated in Table 16. For a given shortcut feature Z associated

Root	Examples	N	P
㉕	so bad , waste your time , not funny , even worse , the worst movie	5,448	1,562
㉕	highly recommended , a must see , a great movie , a great job , a great film	1,484	4,176
⑩	waste of time , bad movie , good thing , terrible movie , horror movie	1,904	297
⑩	must see , great job , great movie , great film , wonderful movie	678	2,530
⑱	at all , at all costs , at least , at best , than this	5,272	2,710
㉛	don ' t waste your time , i mean , don ' t bother , it fails , it was so bad	2,107	463
㉑	worst movie , worst film , worst movies , piece of crap , worst films	2,368	646
㉛	i loved it , i recommend it , i love this movie , i loved this movie , i highly recommend it	191	1,377
⑱	on dvd , as well , in love , at the same time , for everyone	1,471	3,337
⑮	your time , your money , all costs , the worst movies , this crap	7,836	5,567
⑮	the same time , all ages , the best movies , the best , the show	2,413	4,383
㉚	well - , must - , heart - , fun , , fun and	647	2,039
㉚	really bad , boring , , dull , , low budget , so -	2,700	1,069
⑯	3 / , 4 / , 2 / , 1 / , 1 out of	1,673	450
⑨	don ' t , i ' m , there was , the acting is , i could	9,203	7,415
⑦	loved it , love this movie , loved this movie , recommend it , enjoyed it	906	2,229
㉔	of time , of crap , of the worst movies , of my life , of the worst films	3,306	1,692
⑰	at all . , at all costs . , instead . , whatsoever . , ? ? ?	2,333	976
㉙	10 / 10 , 8 / 10 , 7 / 10 , highly recommended . , 9 / 10	5	481
⑦	be funny , work with , sit through , be a comedy , waste your time	1,525	468
⑲	the acting , this movie , the plot , the script , it just	5,849	3,993
⑯	10 / , 8 / , 7 / , 9 / , 7 out of	396	1,304
⑬	bad acting , bad movies , special effects , poor acting , terrible acting	1,623	572
⓪	don ' , couldn ' , didn ' , wasn ' , can '	6,426	4,611
⑤	walter matthau , james stewart , jon voight , william powell , philo vance	85	666
㉙	4 / 10 , 3 / 10 , 1 / 10 , 2 / 10 , 1 / 2 from * * * *	416	13
⑧	30 minutes , five minutes , 10 minutes , 90 minutes , 2 hours	1,083	329
㉑	same time , best movies , first time , best movie , best film	288	966
㉔	of life , of the best , of the best movies , of fun , of my favorites	579	1,424
⑨	it is , i highly , i first , you will , this is	5,595	6,748
①	my favorite , his best , today ' s , my only , my all time	655	1,386
⑰	together . , as well . , today . , very well . , too .	392	978
㉘	' m , ' re , ' t , ' d , ' s	3,283	2,200
㉒	avoid this movie , first of all , i have ever seen , save your money , skip this one	689	248
⑬	great performances , great acting , excellent performances , twists and turns , great fun	100	410
⑭	and enjoy , and sad , 10 / 10 , , as always , worth watching	47	277
⑪	" film , " movie , " plot , " comedy , " so bad it ' s good	207	19
⑤	ed wood , steven seagal , van damme , uwe boll , tom savini	167	7
⑭	or something , . . . , and boring , , right , and pointless	989	501
⑥	. . , . . . , . . well , . . no , . . oh	2,753	1,935
⑲	this game , the series , my only complaint , the film , it also	1,435	2,024
①	their right , your time or , someone ' s , your time and , your local	174	31
㉓	it off , me wrong , it up , through the whole thing , down the toilet	465	205
㉒	a must see , highly recommended , as always , i think , of course	504	793
㉓	out on dvd , me away , - on	12	81
⑳	' n	6	29
⑧	many years	29	66
⓪	you don '	212	282

Table 11: Additional IMDb features (see Section D.1). We report the number of *positive* (**P**) and *negative* (**N**) training examples associated with each feature and highlight the features according to the most common class. Many features are related to clear sentiment markers like adjectives, but it is also easy to identify features corresponding to numerical ratings and other patterns, like actor names, that we might not expect to be correlated with class labels.

Root	Examples	S	O
㉗	his life , his wife , his father , his mother , their lives	323	1,305
㉗	a movie , the film , the movie , this movie , the screen	846	134
⑬	the movie , but it , the film , if you , if it	624	78
㉘	decides to , order to , " " , has been , begins to	133	614
②	" " , best friend , young man , young [UNK] , [UNK] girl	48	423
③	finds himself , finds out , falls in love , is [UNK] , is sent	57	403
㉘	' t , ' s , ' re , ' ll , ' s not	1,396	753
②	[UNK] movie , [UNK] film , running time , romantic comedy , [UNK] plot	279	14
⑬	the two , when he , where he , the gang , the girls	16	234
①	. . , as [UNK] , , too , in a way , in the right place	359	85
⑧	[UNK] [UNK] , [UNK] [UNK] , [UNK] [UNK] , [UNK] [UNK] [UNK] , ' s mother	22	204
①	in love , " " , with him , with her , for her	61	291
③	comes off , ' s hard , makes up , ' d expect , doesn ' t	176	29
㉛	. . . , . ' , of life . , in its [UNK] . , in years .	203	50
⑥	' s] , ' ve seen , ' s also , ' t [UNK] , ' t seen	160	34
⑧	- - , ' s film , and [UNK] , or [UNK] , - [UNK] [UNK]	428	209
⑫	of the film , of a movie , of the year , of a [UNK] , of the plot	82	5
⑫	of his father , of their own , of his life , of the world , of the [UNK]	31	128
㉕	one day , he is [UNK] , along the way , at the same time , in the meantime	1	40
㉕	it ' s [UNK] , it ' s , that ' s , for the most part , the film [UNK]	33	1
⑥	" " [UNK] , ' s got , order to [UNK] , struggle to find , " " tells	5	43
㉛	" " . , on him . , of it . , for [UNK] . , in the [UNK] .	2	34
㉙	the [UNK] , her [UNK] , his [UNK] , their [UNK] , two [UNK]	61	112
㉙	its [UNK] , this [UNK]	27	5
⑮	sang - woo , daniel [UNK] , played by [UNK]	0	12
㉓	[UNK] [UNK] , - and	33	13
⑩	- [UNK] [UNK]	6	1
㉒	the most part	4	0
⑪	silence of the lambs	4	0
⓪	[UNK] of a movie	4	0
㉑	[UNK] [UNK] ,	0	4
㊿	daniel [UNK]	0	4
㉓	year - old	0	4
㉒	this " "	0	4
⑱	& # 214	0	4
⓪	[UNK] of [UNK] [UNK]	0	4
⑪	death of his father	0	3
㉖	dickens '	3	0

Table 12: Additional SUBJ features (see Section D.1). We report the number of *subjective* (**S**) and *objective* (**O**) training examples associated with each feature and highlight the features according to the most common class. These features reflect how this dataset was constructed (Pang and Lee, 2004): the *subjective* class consists of movie reviews from Rotten Tomatoes and the *objective* class consists of movie summaries from IMDb.

Root	Examples	E	C	N
⑭	a man/a woman , a woman/a man , a man/ nobody , a boy/a girl , a dog/a cat	235	1,690	307
⑰	ϵ/to work , ϵ/to get , ϵ/to buy , ϵ/the park , ϵ/on vacation	252	467	1,715
⑭	a man/a person , a man/a man , a woman/a person , man/a man , a man/a human	6,212	3,354	3,833
⓪	in a/on the , on a/in a , on the/in the , on a/in the , on the/in a	555	1,810	793
④	a /there is , ϵ/there are , two /there are , a /there are , ϵ/there is	1,356	449	402
⓪	on a/on a , at a/at a , in a/in a , on the/on the , in a/wearing a	3,074	1,703	1,696
⑰	ϵ/at home , ϵ/in bed , ϵ/' t , ϵ/ice cream , ϵ/watching tv	56	597	210
②	in the grass/ outside , down the street/ outside , in the snow/ outside , on the sidewalk/ outside , in the snow/in the snow	689	107	230
⓪	on a/for a , in a/for a , on the/for a , in the/ a , of a/for a	157	222	752
⑭	a man/the man , a man/a tall human , a man/a couple , a woman/the woman , a woman/a tall human	1,214	1,765	2,375
㉔	black dog/ cat , little girl/a boy , young woman/a man , man/naked man , brown dog/ cat	34	278	35
㉔	man/tall human , woman/tall human , man/old man , man/tall person , man/construction worker	83	135	368
㉔	group of people/there are people , little boy/a boy , black dog/ animal , group of people/several people , young boy/a child	836	367	446
⑯	in the/ϵ , on a/ϵ , down a/ϵ , through a/ϵ , through the/ϵ	9,099	7,829	7,798
⑩	towards the camera/ϵ , in the sand/ϵ , down a road/ϵ , with red - hair/ϵ , on the street/ϵ	1,048	621	573
⑲	in the snow/ outside , sitting on a bench/sitting on a bench , in the snow/ outdoors , in the air/ jumping , in the grass/ outside	78	2	5
㉑	in front/in front , wearing hats/wearing hats , wearing glasses/wearing glasses , upside down/upside down , wearing a hat/wearing a hat	97	12	21
⑩	in his hands/ϵ , ' s hair/ϵ , wearing a brown/ϵ , with a bag/ϵ , on the road/ϵ	28	86	20
⑪	- hair/ϵ , green shirt/ϵ , striped shirt/ϵ , street vendor/ϵ , crowded street/ϵ	160	81	64
⑤	in front/ϵ , and shorts/ϵ , and smiling/ϵ , while people/ϵ , near water/ϵ	572	396	383
④	two young/the , two/the two , four young/the , ϵ/one of , ϵ/a man and	91	147	204
②	down the street/ home , in the sand/on the beach , at a table/ lunch , on a bench/in a park	1	1	28
⑰	ϵ/at least , ϵ/a woman , ϵ/and child	39	4	14
⑯	climb a/ϵ , playing a/ϵ , over their/ϵ , "/ϵ , into the/ϵ	249	304	180
⑪	brown jacket/ϵ , blue plaid/ϵ , blue dress/ϵ , baseball uniform/ϵ , dark shirt/ϵ	17	10	48
㉗	of people/ person , of people/ is , of men/of women	28	81	48
㉛	is walking/is walking , is smiling/is smiling , are dancing/are dancing	21	1	2
④	two /there is , two /there is only , three/the three , three young/the , two young/two old	21	67	38
㉗	of people/are people , of people/of people	210	138	129
⑧	laughing /laughing at , crying /crying because	0	0	14
⑩	down a city/ϵ , in an office/ϵ , and hard hats/ϵ , wearing a black/ϵ	31	20	60
㉗	of people/of friends , of dogs/ dogs	3	0	17
⑤	wearing glasses/ϵ , around her/ϵ , in red/ϵ , of corn/ϵ , of volleyball/ϵ	71	128	95
②	at night/during the day	0	8	0
⑯	on two/ϵ , in red/ϵ , and hard/ϵ	56	67	92
⑤	with black/ϵ , of young/ϵ	38	28	54
⑪	brown hat/ϵ	0	7	1
⑮	soccer ball/playing soccer	13	3	7
⑮	tennis ball/playing fetch	0	0	5
⑥	man and a/ϵ	2	8	1
⑥	man ' s/ϵ	9	2	2
㉑	playing soccer/playing soccer	11	4	14
⑥	side of a/ϵ	17	11	25
㉑	hanging out/hanging out	0	6	2

Table 13: Additional SNLI features (see Section D.1). We report the number of *entailment* (**E**), *contradiction* (**C**), and *netural* (**N**) training examples associated with each feature and highlight the features according to the most common class. *Contradiction* features tend to involve antonyms, *neutral* features tend to involve additions, and *entailment* features involve copied clauses and hypernyms.

Root	Examples	N	P
㉕	new year/new year , world war/world war , donald trump/donald trump	538	2,813
⑭	how can/how can , how do/how can , how can/how do	4,072	7,952
㉛	improve my/improve my , earn money/earn money , make money/make money	622	2,529
㉗	candy imported/candy imported , lose weight/lose weight , writing skills/writing skills	93	1,257
⑭	is /what is , is /how do , is /what are	1,759	410
③	candy imported in/candy imported in , not be/not be , traffic to/traffic on	76	839
⑩	i improve my/i improve my , you have/you have , i earn money/i earn money	316	1,336
⑦	saltwater taffy/saltwater taffy , way to/way to , purpose of/purpose of	251	1,097
㉔	do to/ϵ , to learn/to learn , is //ϵ	231	897
④	ϵ/do to , ϵ/way to , ϵ/and why	742	1,673
③	[UNK] ./[UNK] . , mean in/mean in , politics and/politics and	349	16
②	ϵ/it like , ϵ/" " , ϵ/like to	642	154
④	ϵ/" " , ϵ/' t , ϵ/in [UNK]	740	218
㉔	" "/ϵ , [UNK] in/ϵ , [UNK] [UNK]/ϵ	558	127
㉕	the word/the word , the lewis/the lewis , a sentence/a sentence	335	41
⑲	hollywood movies/hollywood movies , day of your life/day of your life , company in delhi/company in delhi	5	158
㉗	[UNK] . com/[UNK] . com , politics and government/politics and government , blood pressure/blood pressure	128	1
②	ϵ/? what are , ϵ/? what , ϵ/if yes	74	311
⑩	the word '/the word ' , you determine the lewis/ is the lewis , i watch/i watch	103	8
⑧	ϵ/? what are some examples , ϵ/from your perspective , ϵ/? how do they	18	108
⑲	[UNK] . com/[UNK] . com , college in singapore/college in singapore	52	0
⑱	time travel to/time travel , life ?/life , spotify is/spotify	0	49
⑪	will win/will win , i can/ do , music do/music do	8	75
⑤	best day of your life/best day of your life , purpose of life/purpose of life , meaning of life/meaning of life	1	53
㉛	solve this/solve this , determine the lewis/is the lewis , calculate the/calculate the	60	3
⑦	review of/review of , kind of/kind of , aspects about/aspects about	515	303
⑧	ϵ/it like to , ϵ/it like to be , ϵ/like to be	43	1
⑪	competitive is/competitive is , much does/much does , business can/business can	92	21
⑮	what is [UNK] . com/what is [UNK] . com , is [UNK] . com legit/is [UNK] . com legit , what is the meaning of marathi word ' [UNK] '/what is the meaning of marathi word ' [UNK] '	38	0
⑨	what is [UNK] . com ?/what is [UNK] . com ? , is [UNK] . com legit ?/is [UNK] . com legit ? , how is the word ' [UNK] ' used in a sentence ?/how is the word ' [UNK] ' used in a sentence ?	38	0
⑥	[UNK] . com/[UNK] . com , [UNK] [UNK]/[UNK] [UNK]	44	3
①	is [UNK] . com/is [UNK] . com , is [UNK] [UNK]/is [UNK] [UNK]	32	0
⑥	the best day of your life/the best day of your life , some examples/some examples , the point of life/the point of life	3	31
⓪	ϵ/from your perspective , , ϵ/we can remain satisfied in , ϵ/wanna ask someone please	0	21
⑫	[UNK] . com make money/[UNK] . com make money , [UNK] . com legit/[UNK] . com legit , the word ' [UNK] ' used in a sentence/the word ' [UNK] ' used in a sentence	21	0
㉑	long distance relationships/long distance relationship , your life ? what happened/your life , long distance relationships work/long distance relationship	0	19
⑳	spotify is not/spotify , new year ' s/new year , your life ? what/your life	0	18
㉖	[UNK] /[UNK] and , [UNK] /" "	18	1
⑬	a person/i	2	16
⓪	ϵ/it like to be	9	0
⑮	what are some examples/what are some examples	3	15
①	are some examples/are some examples	3	15
⑫	trump win/trump win	0	8
⑱	[UNK] [UNK]/[UNK] [UNK]	25	10
⑬	" "/it	7	0
⑤	meaning of marathi word ' [UNK] '/meaning of marathi word ' [UNK] '	6	0

Table 14: Additional QQP features (see Section D.1). We report the number of *non-paraphrase* (**N**) and *paraphrase* (**P**) training examples associated with each feature and highlight the features according to the most common class. The *paraphrase* features tend to correspond to how-to questions (such as how to earn money) or open-ended discussion questions—for example, about the 2016 presidential elections and the meaning of life. The *no paraphrase* features include subtrees reflecting sequences that appear in both questions and differ only in one, uncommon word, which is replaced with the unknown token. Training examples with this feature include "What is instagramtop.com? What is bestmytest.com?", or "How is the word 'wry' used in a sentence? How is the word 'adduce' used in a sentence?" This pattern appears exclusively in non-paraphrase examples.

Production rule	Spans	E	C	N
⑰ → ㊾ ㉟	(17 (49 /to) (35 /work)) (17 (49 /to) (35 /get)) (17 (49 /to) (35 /buy))	1,990	3,774	6,397
② → ⑰ ②	(2 (17 /the toy) (2 between his legs/between his legs)) (2 (17 /a picture) (2 in the snow/in the snow)) (2 (17 /marco polo) (2 in the pool/in the pool))	627	1,495	2,415
② → ⓪ ㉝	(2 (0 at /during the) (83 night/day)) (2 (0 in the/in the) (83 snow/sand)) (2 (0 down a/down a) (83 street/street))	7,704	10,499	10,112
② → ⑧ ⑰	(2 (8 /outside in) (17 /the summer)) (2 (8 /off to) (17 /some friends)) (2 (8 /music on an) (17 /outside stage))	225	463	1,111
⑧ → ② ㊾	(8 (2 having a conversation/ talking) (49 /about)) (8 (2 an instrument/ music) (49 /for))	260	595	1,200
⑲ → ㉘ ②	(19 (78 walking/walking) (2 down the street/in the mall)) (19 (78 jumping/sitting) (2 in the air/in a chair)) (19 (78 running/swimming) (2 through the snow/in a lake))	3,636	5,361	5,251
⑧ → ⑧ ㊾	(8 (8 cheering /cheering for) (49 /their)) (8 (8 walking /walking down) (49 /a))	161	265	772
⑲ → ㊹ ⑩	(19 (44 walking/walking) (10 down the street/)) (19 (44 running/running) (10 through the water/)) (19 (44 singing/singing) (10 into a microphone/))	1,232	543	416
⑧ → ㉑ ㊾	(8 (21 playing soccer/playing soccer) (49 /in)) (8 (21 taking a picture/taking a picture) (49 /of)) (8 (21 for a picture/for a picture) (49 /after))	260	361	949
② → ㊾ ②	(2 (49 /to) (2 on the corner/cross the street)) (2 (49 /while) (2 reading a book/ reading)) (2 (49 /to) (2 down the street/ work))	369	685	1,130
⑲ → ⑧ ⑰	(19 (8 laughing /laughing at) (17 /a joke))	223	414	847
⑲ → ㉑ ⑩	(19 (21 playing soccer/playing soccer) (10 on a field/)) (19 (21 catching a football/catches a football) (10 with both hands/)) (19 (21 be towed/being towed) (10 by a aaa/))	1,690	1,004	797
⑮ → ㉜ ②	(15 (92 cigarette/smoking) (2 in his mouth/ a pipe)) (15 (92 front/sitting) (2 of a building/ down))	957	1,943	1,627
④ → ㊽ ㊸	(4 (89 a/there) (43 /is)) (4 (89 two/there) (43 /are)) (4 (89 a/there) (43 /are))	1,071	503	427

Table 15: The highest ranked binary production rules by mutual information in SNLI, grouped by majority class (**Entailment**, **Contradiction**, or **Neutral**). Each row shows up to three of the highest scoring subtrees that are generated by that rule and have the same majority class label.

ID	Premise	Hypothesis	y	$\hat{y}$
$\boldsymbol{x}_0$	a white dog running down a path	a black dog sitting on a bush	C	C
$\boldsymbol{x}_1$	a white dog running down a path	a black dog running down a path	C	C
$\boldsymbol{x}_2$	a white dog running down a path	a dog running down a path	E	E
$\boldsymbol{x}_3$	a dog running down a white path	a dog running down a path	E	E
$\boldsymbol{x}_4$	a dog running down a path	a black dog running down a path	N	N
$\boldsymbol{x}_5$	a dog running down a white path	a black dog running down a path	N	C

Table 16: A set of manual contrastive edits we create for the "Adjective antonym" feature. y is the intended label (Entailment, Contradiction, or Neutral) and $\hat{y}$ is the prediction of a BERT classifier. See Section D.3.

with majority label $y^{Z=1}$, we start by identifying validation examples (x_0, y_0) for which $Z = 1$ and $y_0 = y^{Z=1}$, and simplifying x_0 by removing other features that support $y^{Z=1}$, getting a new instance x_1 with $y_1 = y^{Z=1}$, for which $Z = 1$ provides the most evidence in favor of the label. Then we make a series of small changes to get examples x', y' with $y' \neq y_z$: first, several control examples with $Z \neq 0$ to verify that our perturbations change the model's behavior as expected in the absence of the feature ($x_{2,\ldots,4}$), and finally an instance with $Z = 1$ (x_5).

We apply this procedure to the "Adjective antonym" feature, illustrated in Table 16. This feature is associated with the pre-terminal label ㉟ and includes productions like white/black and red/blue. Our test is to modify the premise by moving the adjective from the subject noun to the object noun, which changes the label from *contradiction* to *neutral*. We randomly select ten validation examples that contain the feature and have the majority label, contradiction. For each example, we create four control examples and one test example, as in Table 16. The BERT model makes the expected prediction for 40/40 control examples but misclassifies 9/10 test examples, in each case predicting contradiction rather than neutral. This indicates that the model predicts contradiction when the premise and hypothesis contain contradicting adjectives, even if the adjectives describe different entities.