What Can a Model Learn Without the Premise? Explaining Hypothesis-Only Shortcuts in Natural Language Inference

Sai Sharan Nellutla

Abstract

Natural Language Inference (NLI) focuses on predicting whether the relationship between two sentences is one of contradiction, neutrality, or entailment. A good classifier needs to use both sentences, but in many cases the hypothesis alone is enough to infer the label. This paper examines how well classifiers that use only the hypothesis can solve the problem, which words they use to make decisions, and whether those words causally affect the decision. With only TF-IDF features from the hypothesis, logistic regression reaches 64.0% test accuracy, compared to a majority baseline of 34.3%, which is 83% of the gain achieved by a similar model that also sees the premise (70.1%). Most of the information comes from single words: unigrams alone reach 63.9%. The key cues are negations and words such as "sleeping" for contradiction, vague words such as "outdoors" for entailment, and added detail such as "tall" for neutral.

Four explanation methods disagree on the most important words, since SHAP and random forest importances prioritize frequent words while Naive Bayes ratios prioritize rare ones. Removing the five highest-weighted words (nobody, nothing, no, because, sad) from test hypotheses lowers accuracy 0.77 percentage points more than removing frequency-matched random words, and this extra drop grows to 9.1 points at 100 words. On the 36% of test examples where the hypothesis-only classifier makes mistakes, the classifier that also sees the premise reaches only 31.7% accuracy, compared to 91.6% on the rest. These shortcuts transfer poorly to MultiNLI. A significant amount of success on NLI can therefore come from shortcuts in how the data was written, and the choice of explanation method strongly affects which shortcuts are found.

Keywords: natural language inference, annotation artifacts, shortcut learning, explainable AI, SHAP, dataset bias

1. Introduction

Recognizing natural language inference (or "textual entailment") is a major challenge in measuring machine understanding of language [1, 2]. Each data point consists of two sentences: a premise, which represents a certain scenario ("A woman is in the middle of hitting a tennis ball"), and a hypothesis, which represents a statement ("A woman is sleeping"). The model needs to return one of three possible labels: entailment if the hypothesis is entailed by the premise, contradiction if it cannot be true, and neutral if it could be either true or false. In this case, the correct label is contradiction, because it is not possible to be hitting a tennis ball and sleeping at the same time.

This problem seems especially interesting as a benchmark precisely because of its requirement of meaningful reasoning: the model has to compare the two sentences, find the commonalities, and figure out whether one of the statements rules out the other one. Massively annotated datasets, such as the Stanford Natural Language Inference dataset (SNLI) with its over five hundred thousand sentence pairs [2] and the Multi-Genre NLI dataset (MultiNLI) [3], made it possible to train and compare data-intensive neural network models, and accuracy on such

datasets became the standard way of measuring performance on natural language understanding tasks.

The utility of a benchmark hinges upon how high scores are correlated with the capacity it seeks to evaluate. Several papers found that the hypothesis sentence itself is indicative of the label when looked at in isolation [4], [5], [6]. SNLI was constructed by giving crowdsourced workers a premise from an image caption and asking them to come up with three hypotheses: one that is definitely true, one that could be true, and one that is definitely false. Due to time limitations, workers were prone to take shortcuts. To create a false sentence, a quick solution is adding a negation ("Nobody is...") or making people in the picture fall asleep. To come up with a true sentence, a safer option is to generalize it ("A person is outside"). To make a maybe-true sentence, workers would often add some unverifiable information, such as a purpose or emotions. These tendencies are known as annotation artifacts. It means that a model can attain good accuracy while ignoring the premise completely due to annotation artifacts, showing that some elements of what we consider inferential reasoning could just be pattern matching on the hypothesis sentence.

This kind of behavior highlights a larger problem of shortcut learning [7]: models often discover the easiest feature in the training set that correlates with the label but is not actually a representation of the skill it is supposed to learn. Shortcuts lead to overfitting to easy-to-use features of a dataset and make a model fail in situations where the shortcut is not applicable anymore [8], [9]. Therefore, understanding shortcuts is essential for everyone who works on benchmarks, evaluating or deploying language models.

The discovery of such shortcuts presupposes access to the inner mechanisms of a model, one of the goals of XAI. Popular explanation algorithms, including LIME [10] and SHAP [11], attribute scores to each input feature based on their importance in predicting. Even though these algorithms are widely utilized for the auditing of models, there is an increasing number of studies that show that different methods might give contradictory results for the same model [12], [13]; hence, there arises a practical question of how reliable the conclusion reached by means of some explanation method about a particular shortcut is.

This paper is inspired by the author's earlier internship experience with neuron-level explanations, a method developed by Mu and Andreas [14] in which each neuron in a neural NLI model is described by logical rules over input features. Previously, it was found that many neurons in an SNLI model react to certain hypothesis words, such as “nobody”, as well as to verbs “sleeping” and “eating”, rather than to relations between the premise and the hypothesis [14]. The question that naturally arises after this result is related to the data itself: if the units inside a neural network mostly react to these words, then to what extent the task can be solved by using them only, and which explanations can reveal them? The current paper attempts to answer this question by exploring simple interpretable models.

1.1 Research questions

The study has been organized around the following three research questions:

- **RQ1**: To what degree are simple models able to predict NLI labels based solely on the hypothesis? Four classifiers using only the hypothesis have been trained on SNLI data and compared with a majority-class baseline and with a model that uses both sentences.

Moreover, the influence of the training set size, textual features used, and data set (based on MultiNLI evaluation) on the accuracy has been analyzed.

- **RQ2**: Which words play a role in making these predictions, and are there any convergences between the sets of influential words obtained by various models and explanation techniques? Word rankings based on logistic regression coefficients, Naive Bayes probability ratios, SHAP scores, and random forest importance scores have been calculated, and agreement between the rankings has been measured.
- **RQ3**: How meaningful are these words for prediction? Top words have been deleted from test hypotheses, and the effect on accuracy has been compared to the one caused by deletion of words of similar frequency chosen randomly.

1.2 Contributions

Accuracy above chance with a hypothesis-only classifier on the SNLI corpus is well-documented [4], [5]. In addition to replicating the finding with a completely transparent analysis pipeline, the current study makes the following contributions:

- A thorough comparison of four hypothesis-only classifiers together with a learning curve analysis, an n-gram ablation, and a cross-dataset evaluation showing that the shortcut cue is learnable from little data, is mostly conveyed by single words, and is mostly unique to SNLI.
- An easy/hard analysis suggesting that the success of a premise-aware bag-of-words classifier is confined to those examples that are already solved by the hypothesis-only classifier and is below chance otherwise.
- A comparison of four different explanation methods of the same hypothesis-only classifier task revealing that the source of their discrepancies is different word-frequency weighting.
- A careful word removal experiment involving frequency-matched random controls and an additional design change for the controls revealed in the supplementary material to determine if the selected words have a causal impact on predictions.
- A completely reproducible pipeline where all results come from scripts with fixed random seeds and a list of issues and design choices made during the process available for public review.

Figure 1 gives an overview of the pipeline. Related work is discussed in Section 2. Section 3 covers the data, Section 4 covers the methods, and Section 5 the results. Discussion and limitations of the results are presented in Section 6 while conclusions are provided in Section 7. The appendices contain reproducibility information and full result tables.

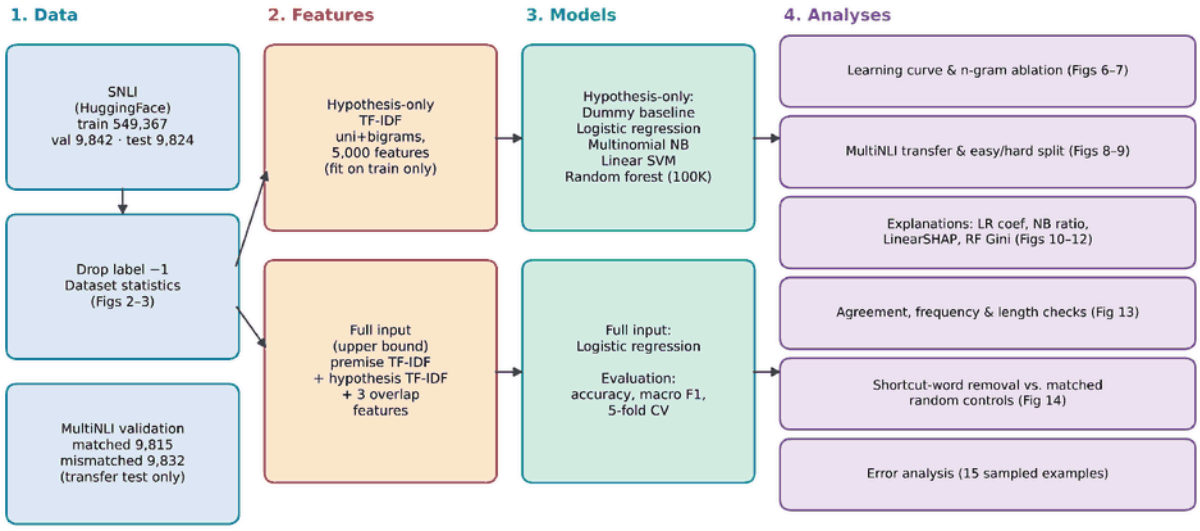


Figure 1. Overview of the study. SNLI is cleaned and summarized (Section 3), converted into hypothesis-only and full-input TF-IDF features (Section 4.2), and used to train five hypothesis-only models and one full-input model (Section 4.3). Six groups of analyses follow (Sections 4.5 to 4.8). MultiNLI is used only to test how well the SNLI-trained models transfer to new data.

2. Background and Related Work

2.1 Natural language inference and its datasets

The modern form of NLI problem is derived from PASCAL Recognizing Textual Entailment series of challenges [1] where the system had to find out whether the text entailed another text. The early datasets contained no more than a few thousands examples. The scale of NLI community was significantly expanded by the creation of the Stanford Natural Language Inference (SNLI) corpus [2]. SNLI contains premises as image captions from Flickr30k dataset and hypotheses generated by workers from Amazon Mechanical Turk crowdsourcing platform. As a result of the fact that every premise was given just one hypothesis per label, these three labels are almost balanced and majority class baseline reaches just about one third of accuracy. A part of the dataset was re-annotated by four additional annotators and instances without the majority label were given a specific label (-1) and usually excluded from analysis.

Multi-Genre Natural Language Inference (MultiNLI) [3] used the same approach to collect the dataset but got the premises not from images but from ten genres of written and spoken English (fiction, government reports, phone conversations and travel guides). It divides its evaluation set into the matched set containing the genres included in the training set and mismatched set containing other genres. Because MultiNLI has longer and more diverse sentences in comparison with SNLI, MultiNLI is commonly used to evaluate generalization ability of models on the visual and every day scenes from SNLI.

2.2 Annotation artifacts and hypothesis-only baselines

Three independent studies conducted in 2018 confirmed that hypothesis-only models exhibit performance significantly better than chance on NLI datasets. Gururangan et al. [4] used a fastText bag-of-words model to classify the hypotheses only and achieved 67.0% accuracy on SNLI and around 53% on MultiNLI. They used the pointwise mutual information technique to extract words highly correlated with each label and found the same patterns as mentioned above: negations for contradiction, generic and approximate words such as "some" and "at least" for entailment, and modifiers, superlatives, and purpose clauses for neutral. They found out that the length of hypotheses depends on the label, with neutral hypotheses being the longest ones. Poliak et al. [5] confirmed that hypothesis-only baselines outperform the majority class on many of the ten NLI datasets analyzed, and Tsuchiya [6] showed similar latent bias in SNLI. These phenomena have been shown since then to exist in other tasks too, namely in question answering models which can answer questions without even reading the passage [15] and in fact-verification models which can make a judgment about the claim without the evidence at all [16].

This work continues the above research. It uses the same kind of bag-of-words features but adds several analyses that earlier papers did not combine: different model families, a learning curve analysis, n-grams ablation, transfer to MultiNLI without retraining, explanation methods comparison, and controlled removal.

2.3 Shortcut learning and robustness

Shortcut learning is defined by Geirhos et al. [7] as a property of machine learning systems: an algorithm tuned to reduce the error will use all features that correlate with labels within the training data regardless of whether or not these are the intended features for humans. In the case of NLI, McCoy et al. [8] built the HANS dataset to test three such shortcuts based on overlap of words within the premise and the hypothesis and show that strong models trained on MultiNLI fail drastically compared to chance on examples that violate these shortcuts. As a solution, Nie et al. [9] introduced the Adversarial NLI dataset wherein the authors crafted examples to trick existing models. Training strategies focused on avoiding reliance on features from the hypothesis only were also suggested, including training a separate hypothesis-only model and punishing the primary model for using information provided by this model [17], [18]. All of these approaches require the identification of the employed shortcuts in the first place, which is what the presented paper strives to achieve.

2.4 Explaining model predictions

Feature-attribution techniques explain the predictions by assigning a contribution score to each input feature. The Local Interpretable Model-agnostic Explanations (LIME) [10] constructs a simple local surrogate model around a single prediction point. The SHAP method [11], inspired by the Shapley values in cooperative game theory [19], distributes the prediction among the features in the unique way that satisfies a minimal set of criteria. For linear models with independent features, the SHAP values have a simple closed form, making them particularly appropriate for the models considered in this paper (Section 4.6). Inherent interpretability techniques, such as the weights of a linear model, the ratio of class probabilities in the Naive Bayes classifier, and the impurity-based importance measure in a random forest [20], generate the explanations themselves.

It is not necessarily the case that an explanation is correct. Attention weights of the models in NLP were found to be irrelevant for the predictions by Jain and Wallace [21]. Some saliency methods can produce the same explanations for a model with randomized weights [13]. The disagreement issue was documented by Krishna et al. [12]: different methods give conflicting rankings of features even when applied to the same model, while there is no principled way to pick one. A common approach to evaluation of an explanation is the removal experiment: removing the features identified by the explanation as relevant and measuring the drop in model performance. Hooker et al. [22] introduced this type of experiment and stressed its importance in comparison with the random baseline. The removal experiment in Section 4.7 employs this idea with the frequency-matched control group.

2.5 Neuron-level explanations

Another research direction looks into the particular content recognized by the neurons inside the neural network. Network Dissection [23] mapped neurons in an image classification network against human-annotated visual concepts. More recently, Mu and Andreas [14] took this idea further by developing a system that generated compositional explanations, which defined neuron behavior as a logic formula in terms of input features such as "hypothesis contains nobody AND premise does not contain hair." Applying this to a recurrent neural network trained on SNLI, the researchers found many neurons that implemented simple lexical rules associated with dataset biases, including reactions to the word "nobody" and verbs such as "sleeping," "eating," and "sitting" in the hypothesis, as well as high word overlap. In particular, NLI neurons, whose behaviors could be explained by such simple rules, exhibited low accuracy, which means that those rules are learning spurious correlations and not real inference [14].

The neuron-level findings described above tell us what the neural network learns, but not why this information is learnable at all. In contrast, the current work takes the complementary, data-centric approach, quantifying the amount of label information carried by individual hypothesis words in SNLI, through simple enough models so that the answer is immediately visible from the results.

3. Data

3.1 SNLI

All key experiments use SNLI [2] from the HuggingFace Datasets library [24], which is accessible via the identifier stanfordnlp/snli. The dataset consists of standard train, validation, and test splits, which have 550,152, 10,000, and 10,000 examples, respectively. Examples with the gold label of −1, where the majority of annotators did not agree, were filtered out from all splits: 785 from the training split, 158 from the validation split, and 176 from the test split. It gives 549,367 training, 9,842 validation, and 9,824 test examples. Models were trained on the training split and tested on the test split. The validation split was used only for providing statistics about the dataset since there was no tuning of hyperparameters.

3.2 MultiNLI

In order to evaluate the ability of patterns discovered on SNLI to generalize to other domain of texts, we also use two splits of MultiNLI [3], accessible through the nyu-mll/multi_nli identifier. The matched split has 9,815 examples and the mismatched split 9,832, none of which has an

example with label −1. MultiNLI was not used for training: SNLI-trained models and feature extractors were used with it without any changes (Section 4.5).

3.3 Exploratory analysis

The cleaned splits are provided in Table 1. The three labels are almost evenly distributed in all splits, taking 32.8%−34.3% of the examples. Entailment is the most common label in the test split (34.3%), so that a classifier predicting entailment always will obtain 34.3% accuracy in it. This result matches the majority baseline reported by Gururangan et al. [4], suggesting that our cleaned split is equivalent to the standard one.

Table 1. SNLI examples and hypothesis length by split and label, after removing label −1. Length is the number of whitespace-separated words in the hypothesis.

Split	Label	Examples	% of split	Mean length	Median length
train	entailment	183,416	33.4%	6.63	6
train	neutral	182,764	33.3%	8.26	8
train	contradiction	183,187	33.4%	7.36	7
train	ALL	549,367	100.0%	7.42	7
validation	entailment	3,329	33.8%	6.81	6
validation	neutral	3,235	32.9%	8.35	8
validation	contradiction	3,278	33.3%	7.40	7
validation	ALL	9,842	100.0%	7.51	7
test	entailment	3,368	34.3%	6.84	6
test	neutral	3,219	32.8%	8.35	8
test	contradiction	3,237	33.0%	7.29	7
test	ALL	9,824	100.0%	7.48	7

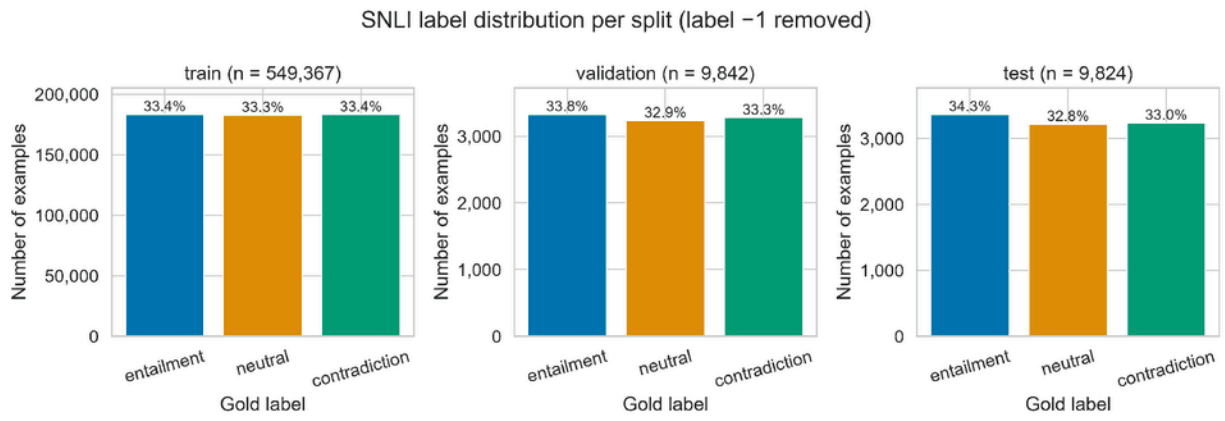


Figure 2. Label distribution in each SNLI split after removing examples with label −1. The near-perfect balance means that any accuracy far above one third must come from information in the input.

The length of the hypothesis is determined according to the label (refer to Table 1 and Figure 3). In the training set, for entailed hypotheses, the length is 6.63 words, while the contradicting hypotheses' length is 7.36 words, and the neutral hypotheses' length is 8.26 words; the same order was seen in both the validation and test sets. This matches well with the annotation process explained in Section 1, in that the entailed hypotheses tend to be shorter since they provide more general statements about the premise. On the other hand, the neutral hypotheses add some information that is not provided in the premise, which makes them longer.

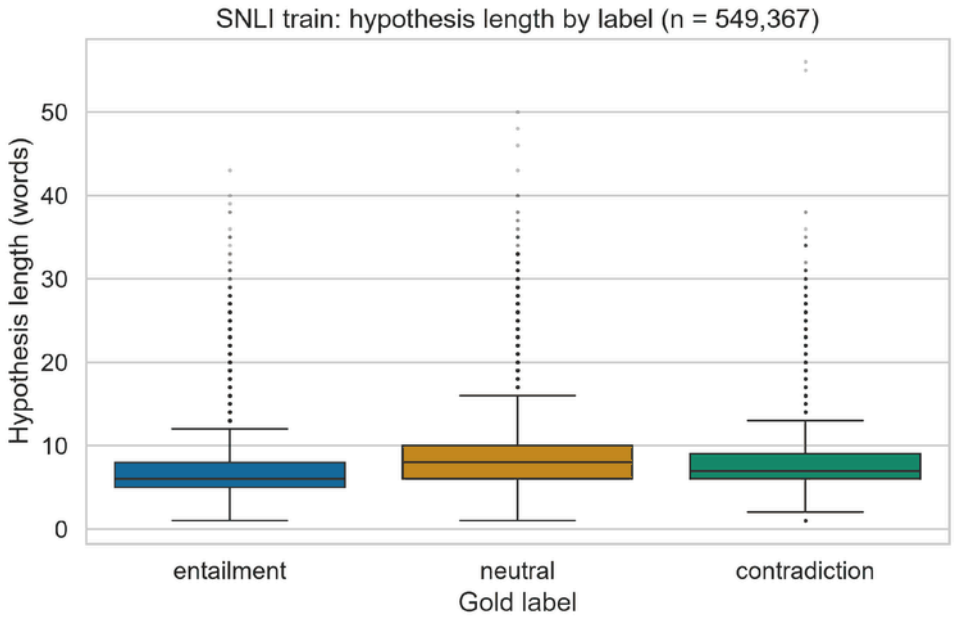


Figure 3. Hypothesis length in words by gold label (SNLI training split). Boxes show the interquartile range and the line marks the median. Neutral hypotheses are longest and entailed hypotheses shortest.

Table 2 provides the most frequent terms that correspond to each label's hypotheses, with English stopwords excluded but negations retained. The top part of the distribution is made up of commonly occurring terms like "man," "woman," "people," and "playing," proving that high term frequency does not indicate the predictive power for a label. However, there are still some distinctive label-specific terms. Namely, "Outside," "person," "outdoors," and "near" occur more often in entailments; "young," "waiting," and "children" in neutrals; and "sleeping," "eating," "running," and "riding" in contradictions. Predictive but low-frequency terms, including negations, are not among the frequencies provided and have to be accounted for through models and explanation techniques presented further below.

Table 2. The 15 most frequent hypothesis words for each label in the SNLI training set. The numbers in parentheses represent the number of times a term appears per 1,000 hypotheses. (The full list of 20 terms can be found on the project's webpage; the token "s" present in the neutral list is a result of breaking down possessives like "man's.")

Rank	Entailment	Neutral	Contradiction
1	man (204)	man (207)	man (213)
2	people (146)	woman (101)	woman (118)
3	woman (101)	people (95)	people (92)
4	outside (80)	playing (51)	playing (55)
5	person (56)	men (50)	sitting (54)
6	playing (54)	boy (47)	men (53)
7	men (48)	girl (47)	girl (49)
8	girl (43)	dog (44)	boy (47)
9	wearing (43)	wearing (34)	dog (45)
10	boy (42)	group (29)	wearing (37)
11	dog (40)	young (29)	sleeping (32)
12	group (31)	women (28)	women (32)
13	sitting (31)	person (27)	group (29)
14	child (30)	outside (27)	person (27)
15	standing (29)	child (26)	young (26)

4. Methods

4.1 Overview and general setup

All experiments follow the same methodology, which can be summarized by providing a model with a numeric encoding of the text and training it to predict the label, after which the model is analyzed. All code was written in Python and makes use of scikit-learn [25] and SHAP [11]. All computations were done on a Windows 11 laptop with 13th-gen Intel Core i7 CPU (12 logical cores), 16GB of RAM, and without a GPU. For all stochastic processes, namely data sampling, model training, SHAP sampling, and plotting, the random seed was fixed at 42. Each feature extractor and scaler was only fitted on the train set to avoid leaking information from the test set into the training process. Precise version numbers of used packages and the order of script execution are documented in Appendix A.

4.2 Text representation

In contrast to the data fed to machine learning models, which is always numerical, all hypotheses were converted to vectors through the TF-IDF (term frequency – inverse document frequency) method [26]. The first step in the TF-IDF procedure is building a term vocabulary. Each sentence is then presented as a vector with one component per vocabulary term: if the

term is not present in the sentence, the corresponding value is zero, otherwise, it depends on how frequently the term appears in the sentence and how infrequently the term appears in the training set. Thus, common function words like "a" get low weights, while rare distinctive terms get high weights. The vocabulary contains both unigrams (like "sleeping") and bigrams (like "nobody is"), is lower-cased, and consists of the 5,000 most frequent terms in the training hypotheses. Since the tokenizer provided by Scikit-learn by default ignores one-character tokens, terms like "a" are not features at all. This representation is called the bag-of-words model because of the order-blind nature of the encoding (though adjacent pairs are retained).

Hypothesis-only features. All primary models use only the TF-IDF vector of the hypothesis and ignore any other data.

Full-input features. In order to estimate the role of the premise for the task, a comparison model uses three feature blocks: TF-IDF vectors of the premise and the hypothesis built using separate vocabularies under the same conditions; and three overlap features: (i) the proportion of hypothesis terms appearing in the premise, (ii) the number of common terms, and (iii) the difference between hypothesis length and premise length. These last three features were normalized to have zero mean and unit variance based on the statistics in the training set. This model is not an NLI system achieving the state-of-the-art results; rather, it is a simple upper bound on the quality reachable with such type of features if both sentences are available.

4.3 Models

In addition to a full-input model, five models based on hypothesis features only were trained, representing different classifiers.

- **Dummy classifier.** As a baseline, the classifier always predicts the most frequent label in the training set, ignoring the input and thus providing a measure of the accuracy that can be achieved by random guessing.
- **Logistic regression** (LR). A linear model calculating a weighted sum of the features per label and transforming the sums into probabilities for the multinomial distribution over three labels. Each weight, or coefficient, represents how much the particular term affects the probability of the given label. L-BFGS solver was used by default, with maximum number of iterations equal to 1,000. The multinomial setting for three labels was used.
- **Multinomial Naive Bayes (NB) [27]**. Probabilistic classifier estimating the probability of a term appearing for each label and assuming conditional independence of terms given the label. NB is computationally inexpensive and a tradition of text classification, in addition, it provides another type of explanations through per-label probabilities of the terms.
- **Linear support vector machine (SVM) [28]**. Linear classifier similar to the LR but aimed to maximize the margin between the labels. SVM was included to see if the results depend on the objective of the LR.
- **Random forest (RF) [20]**. An ensemble consisting of 200 decision trees grown on a random subset of the training set with a random feature selection and making decisions by majority vote. Contrary to the linear classifiers, a forest is able to learn interactions between the terms: for example, the importance of a particular term depending on the absence of some other term. Because of computational restrictions, growing 200 fully-grown trees on more than half a million examples turned out to be impossible.

Hence, the forest was trained on a stratified random subset of 100,000 training examples (33,387 entailment, 33,268 neutral, 33,345 contradiction).

Full-input model is a logistic regression trained on full-input features and using the same parameters as LR trained on hypothesis features. Other hyperparameters are scikit-learn defaults, since the goal is to evaluate what do the untuned standard models learn, not to maximize the accuracy.

4.4 Model Evaluation

Models were tested on the SNLI dataset. The main performance measure was accuracy, the ratio of the correctly labeled examples. Taking into account the possibility of achieving good accuracy through bias to a certain label, the study provides the result in macro F1, the average of F1 scores across the three classes, with F1 being a combination of precision (the ratio of correct examples of a particular class among the predictions for this class) and recall (the ratio of true examples of a particular class). In order to make sure the results were not dependent on one data split, the three fast hypothesis-only models (logistic regression, Naïve Bayes, and support vector machine) were also evaluated using the stratified five-fold cross-validation on the same sample of 100,000 examples used for the forests. In each fold, the TF-IDF vocabulary was rebuilt based on the training part of this fold. Time of training was measured as wall-clock time spent on fitting the model.

4.5 Other accuracy experiments

Learning curve. In order to estimate the amount of data required for learning the shortcuts, hypothesis-only logistic regression was trained on stratified samples with the number of training examples equal to 1,000, 5,000, 10,000, 50,000, 100,000, and the full training set, with the TF-IDF vocabulary rebuilt for each sample. For the samples of 10,000 and less, there were three random seeds (42, 43, 44), and the average with the standard deviation is provided.

N-gram ablation. In order to determine which type of term carries the label signal, hypothesis-only logistic regression was trained using three vocabularies of 5,000 terms: only unigrams, only bigrams, and both (default).

Transfer to MultiNLI. The hypothesis-only models (logistic regression, Naïve Bayes, and SVM) and the full-input logistic regression model were used on the MultiNLI dataset for matched and mismatched subsets without additional training, using the pre-computed feature extractors on SNLI. The SNLI-trained dummy classifier (which always predicts entailment) was used as the baseline; entailment is also the majority class on both subsets. Since the baselines in these subsets and in SNLI are different, the fraction of the gain from the SNLI baseline is reported.

Easy/hard split. Following Gururangan et al. [4], each SNLI test example was marked as easy, if the hypothesis-only logistic regression predicted it correctly, and hard otherwise. Then, the full-input logistic regression model was tested on the two subsets.

4.6 Explanation methods

A total of four approaches have been applied to rank terms according to their informativeness of each label. In any of the four approaches, the ranking of a label is the ordering of terms according to their tendency to bias the prediction toward the label.

1. **LR coefficients**. The coefficient of term j of label c in the hypothesis-only LR. This is a large positive number if the presence of term j substantially increases the score of label c irrespective of the term frequency.
2. **NB log-probability ratio.** For label c, this is the difference between the logarithm of the probability of term j under label c and its logarithm under the other two labels, estimated in the fitted NB model with the label-specific smoothing. Similar to the LR coefficient, it expresses the degree of influence of one occurrence of a term on the label.
3. **LR SHAP.** The SHAP [11] values of the terms in the hypothesis-only LR, calculated using the exact linear explainer on a fixed sample of 1,000 random test examples. For a linear model, the SHAP value of term j in example i is equal to the product of the coefficient and the difference between the feature value of the example and the training-set average of the feature value. This has been checked numerically. The importance of term j of label c is defined as the average of the absolute SHAP values of the term in the sample with the sign of the average effect of the term.
4. **RF Gini importance.** This is the inbuilt importance measure of each term in the random forest that is the average of the reduction of impurity due to the splits on that term across all trees [20]. SHAP values for the random forest were also supposed to be calculated, but the precise tree SHAP could not produce even one example in eight minutes due to the fact that 200 complete trees contain an average of 39,000 leaves each; a 1,000-example run would take many days. Thus, random forest SHAP was abandoned, and Gini importance became its proxy (Appendix D).

Agreement. Top 100 terms for each label produced by the three label-specific methods (LR coefficients, NB ratios, LR SHAP) were combined, resulting in a pool of 219 to 237 terms, and the Spearman rank correlation coefficient ρ [29] between each of the three pairs of methods was calculated over that pool. The value of Spearman's ρ ranges from +1 (same order) via 0 (no relation) to −1 (reverse order). Since Gini importance does not depend on label, it was compared with each other method's average importance magnitude across labels based on the union of the four top-100 lists (276 terms).

Checks for frequency and length. To help figure out possible reasons for disagreements between methods, two additional checks were performed. Firstly, for each label and each method, the Spearman's ρ was calculated between the method's score for the term and its document frequency, which is the number of training hypotheses containing that term. Secondly, to check the possibility that frequent function words become a proxy of sentence length, Spearman's ρ was calculated between SHAP value of each of the ten most highly ranked SHAP terms on the neutral label and hypothesis length, using the same 1,000-example sample.

4.7 Shortcut removal experiment

It is only valuable to explain the importance of words that have an impact on the model. This is evaluated by removing the top k words from all test hypotheses for k = 5, 10, 20, 50 and 100, converting the edited hypotheses to features using the original fitted vectorizer, and evaluating the hypothesis-only LR with its original parameters. Removing a word means removing all bigrams that contain it. Two rankings are considered, each limited to unigrams:

1. **LR SHAP ranking**: words ranked according to their maximum SHAP importance for any label.
2. **LR coefficient ranking**: words ranked according to their maximum absolute coefficient for any label, restricted to words occurring in at least 0.1% of training hypotheses (at least 550 out of 549,367). This makes the ranking less likely to include rare words which would change almost none of test hypotheses if deleted.

Deleting words will decrease the accuracy to some extent, simply because some information is lost. The effect of deleting shortcut words is compared to a randomly selected alternative. There are two designs for this control: one groups words in 20 frequency bins, and replaces words with words drawn from the same bin; another matches each shortcut word to the non-shortcut word closest in frequency rank, excluding the top 100 words from the ranking. The former control proved to be not well-matched to very frequent words because of the large span of frequencies in the highest bin. It was redesigned after preliminary results revealed this issue – now each word in the ranking is matched to a word from outside the ranking in the same frequency rank, excepting the top 100 words, with five different seeds (from 42 to 46) determining the order of matching. The former control is described in the appendix C to ensure full transparency about changes in the analysis. For each condition the average training document frequency of the replaced words and the number of changed test hypotheses is provided.

4.8 Error analysis

To support the overall statistics with some real-world examples, 15 samples have been selected randomly (seed=42) from the following categories: (A) hypothesis-only LR is right with a confidence higher than 0.9; (B) hypothesis-only LR is wrong with a confidence higher than 0.9; and (C) hypothesis-only LR is wrong, while full-input LR is right. Here, the confidence means the highest probability of predicted class. Random selection of samples is preferred over selecting only the most extreme samples, to avoid cherry-picking. In each sample, three terms with highest SHAP values for the hypothesis-only model are included.

5. Results

5.1 How well do hypothesis-only models perform?

Table 3 and Figure 4 answer the first research question. Every hypothesis-only model scores far above the dummy baseline of 34.3%. Linear SVM (64.06%) and logistic regression (64.03%) are essentially tied as the best hypothesis-only models, followed by Naive Bayes (61.42%) and the random forest (58.65%). Macro F1 tracks accuracy closely for all trained models, which means the accuracy is not coming from favoring one label. The dummy's macro F1 of 0.17, in contrast, shows that it predicts only one label.

Table 3. SNLI test results (n = 9,824). "Train time" is the wall-clock time of the fitting step only, in seconds. The random forest was trained on a 100,000-example subsample.

Model	Input	Training examples	Accuracy	Macro F1	Train time (s)
Dummy (most frequent)	Hypothesis	549,367	0.3428	0.1702	0.02
LogReg (hyp-only)	Hypothesis	549,367	0.6403	0.6398	18.57

Model	Input	Training examples	Accuracy	Macro F1	Train time (s)
MultinomialNB (hyp-only)	Hypothesis	549,367	0.6142	0.6139	0.09
LinearSVC (hyp-only)	Hypothesis	549,367	0.6406	0.6400	92.64
RandomForest (hyp-only, 100k)	Hypothesis	100,000	0.5865	0.5858	277.25
LogReg (full input)	Premise + hypothesis	549,367	0.7005	0.6993	73.34

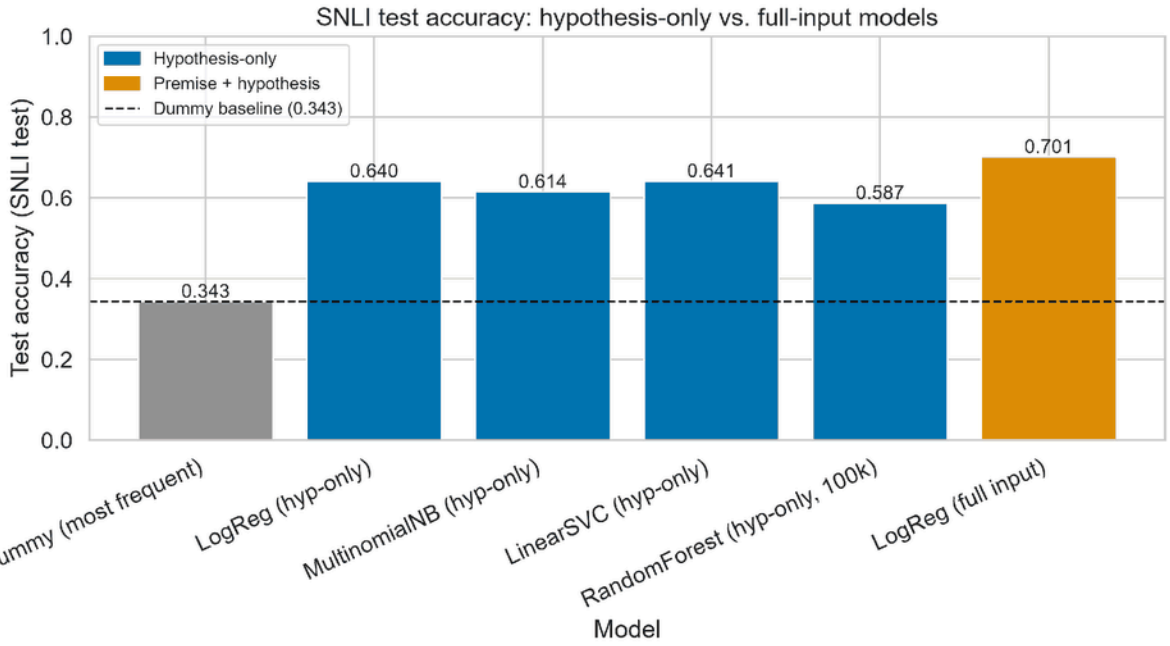


Figure 4. Test accuracy of every model. Blue bars are hypothesis-only models, the orange bar is the premise-aware model, and the dashed line marks the dummy baseline (0.343).

The premise-aware model, which inputs both sentences and three overlap features, scores 70.05%. This is just 6.0 percentage points better than the best hypothesis-only model. In other words, the hypothesis-only LR gets 83% of the improvement of the full model over baseline (29.8 out of 35.8 points) without using the premise at all. For the bag-of-words feature family, then, the premise provides little extra information; the label is mostly contained in the hypothesis. The slight improvement also reflects a limitation of the features themselves: since premise and hypothesis are both represented as bags of words, a linear model can sum the evidence in each sentence but not verify whether a hypothesis word contradicts a premise word except through the three overlap features.

The random forest is the least successful model among those trained. This is partially due to the smaller size of its training set: based on the learning curve presented in Section 5.2, LR trained on 100,000 examples gets 62.8%, leaving the forest 4 points behind even with the same

number of examples. Random forests trained on tens of thousands of sparse TF-IDF features have been shown to be a poor match for text classification tasks because each split of the tree can consider only one term, and the vast majority of terms will be missing in most sentences. Training time varies significantly, from less than a tenth of a second for Naive Bayes to 277 seconds for the forest, but all models are easily trainable on a laptop.

As seen in Table 4, the hypothesis-only models identify all three labels equally well. Entailment has the highest recall for both linear models (0.67 to 0.68), while contradiction has the lowest (0.60 to 0.61). Confusion matrix in Figure 5 demonstrates that the errors made by LR are relatively evenly distributed: 21% of the contradictions are mistakenly classified as entailment and 19% as neutral, and there is no systematic confusion of any label with another label.

Table 4. Per-label precision (P), recall (R) and F1 on the SNLI test split. The dummy classifier is omitted because it only predicts entailment (P = 0.343, R = 1.000 for entailment, 0 for the other labels).

Model	Ent. P	Ent. R	Ent. F1	Neu. P	Neu. R	Neu. F1	Con. P	Con. R	Con. F1
LR (hyp.)	0.643	0.671	0.656	0.646	0.641	0.643	0.632	0.608	0.620
NB (hyp.)	0.616	0.635	0.625	0.612	0.613	0.613	0.615	0.594	0.604
SVM (hyp.)	0.633	0.681	0.656	0.651	0.639	0.645	0.638	0.600	0.618
RF (hyp., 100K)	0.596	0.617	0.606	0.580	0.604	0.592	0.583	0.537	0.559
LR (full)	0.725	0.762	0.743	0.680	0.661	0.670	0.694	0.676	0.684

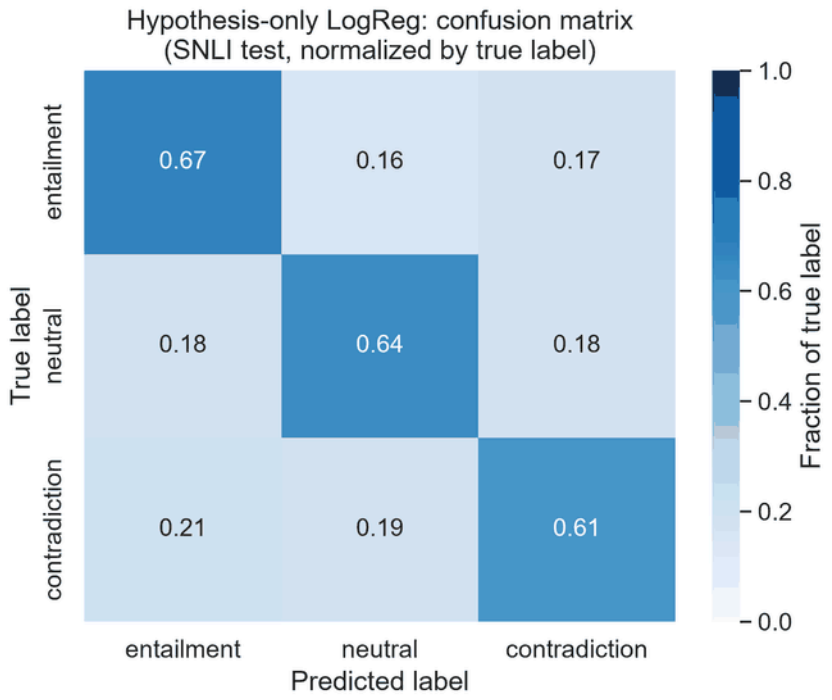


Figure 5. Confusion matrix of the hypothesis-only logistic regression on the SNLI test split, normalized by true label. Each row sums to one; the diagonal shows recall for each label.

The cross validation demonstrates that these findings are reliable (Table 5). For each of the five folds in the subsample of 100,000 examples, the LR shows an average score of 61.57% with a standard deviation of just 0.36 points, and the SVM and NB have almost similar standard deviations. The scores for cross validation are approximately 2.5 percentage points lower than the test scores since each fold learns from 80,000 examples rather than 549,367 examples as seen in the learning curve below. Lastly, the forest model was run again using the same settings to measure time and returned the same 9,824 test predictions as the first run.

Table 5. Five-fold stratified cross-validation accuracy on the 100,000-example training subsample. The TF-IDF vocabulary was rebuilt inside each fold. Standard deviations are population values.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean ± SD
LogReg (hyp-only)	0.6194	0.6101	0.6149	0.6198	0.6143	0.6157 ± 0.0036
MultinomialNB (hyp-only)	0.5975	0.5858	0.5976	0.5996	0.5908	0.5943 ± 0.0052
LinearSVC (hyp-only)	0.6172	0.6081	0.6098	0.6138	0.6104	0.6119 ± 0.0033

5.2 How much data does it take to learn the shortcuts?

The learning curve (Table 6 and Figure 6) demonstrates that the shortcuts can be learned from very little data. It takes 1,000 training hypotheses, which account for less than 0.2% of the whole

training set, for LR to get to 49.2%, 15 points above chance. It gets to 57.5% after 10,000 training hypotheses and to 62.8% after 100,000 training hypotheses, which is less than 1.2 points away from the maximum accuracy achieved using the entire training set. Accuracy continues to increase until reaching the training set size, which means that there is some benefit to having more data, but most of the signal from the shortcuts can be extracted from a tiny subset.

Table 6. Learning curve of hypothesis-only logistic regression on the SNLI test split. Sizes up to 10,000 were repeated with three seeds and show mean ± standard deviation.

Training examples	Seeds	Accuracy	Macro F1
1,000	42 43 44	0.4920 ± 0.0026	0.4917 ± 0.0029
5,000	42 43 44	0.5549 ± 0.0008	0.5545 ± 0.0006
10,000	42 43 44	0.5749 ± 0.0031	0.5745 ± 0.0032
50,000	42	0.6171	0.6166
100,000	42	0.6279	0.6275
549,367	42	0.6403	0.6398

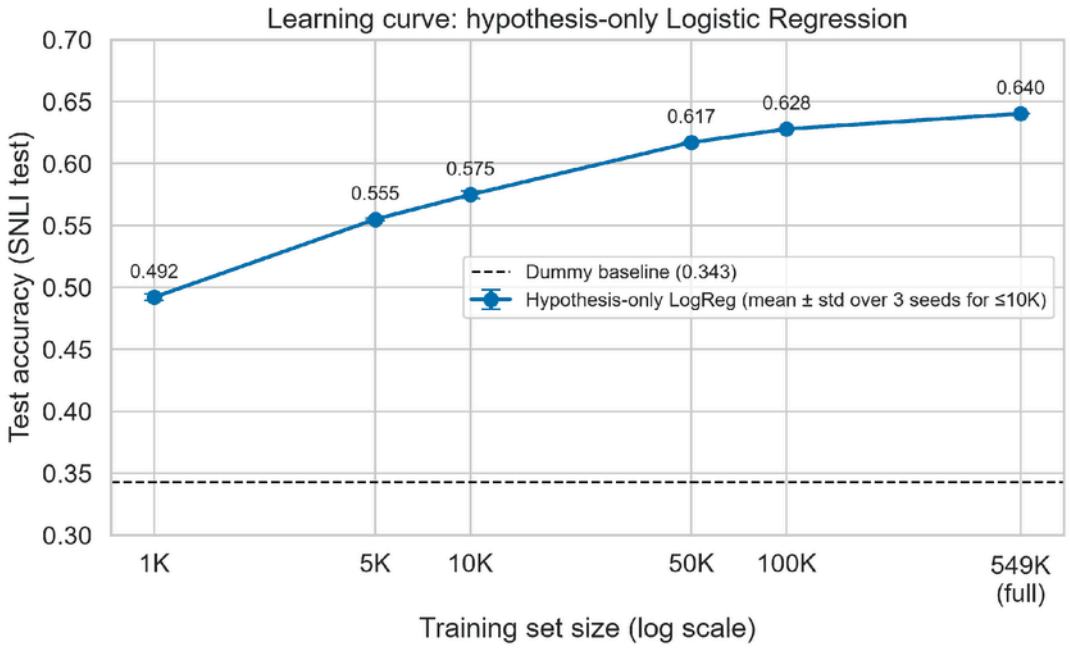


Figure 6. Accuracy of logistic regression on hypothesis only as a function of training set size (logarithmic scale). Error bars show the standard deviation over three different seed values up to 10,000 samples. Dashed line is the dummy baseline.

That makes a difference in analyzing why neural networks tend to use those clues. Something that can be detected by a linear model using just 1,000 training examples is the easiest clue for

a much more complex model to detect, and that is exactly what is expected when shortcut learning happens [7].

5.3 Location of the Signal: Words or Phrases?

N-gram ablation (Table 7 and Figure 7) demonstrates that almost all of the signal comes from single words. Usage of unigrams provides 63.88% accuracy, being just 0.15 points worse than 64.03% of accuracy for using both unigrams and bigrams. Usage of bigrams yields 59.48%, being 4.5 points worse. Some part of this gap is technical: 509 (5.2%) test hypotheses do not have any bigrams in the 5,000 vocabulary, so the bigram-only model cannot process them and makes default predictions only. Nevertheless, the most important conclusion is clear: the shortcuts are mostly properties of single words like "nobody" or "sleeping" rather than long phrases. This is in agreement with neuron-level analysis of SNLI models and word-level concepts found there [14].

Table 7. N-gram ablation for hypothesis-only logistic regression (5,000 TF-IDF features each, full training set). The last column counts test hypotheses with no feature in the vocabulary.

Setting	Accuracy	Macro F1	Empty test inputs
Unigrams (1,1)	0.6388	0.6383	3
Bigrams (2,2)	0.5948	0.5938	509
Uni + bigrams (1,2)	0.6403	0.6398	7

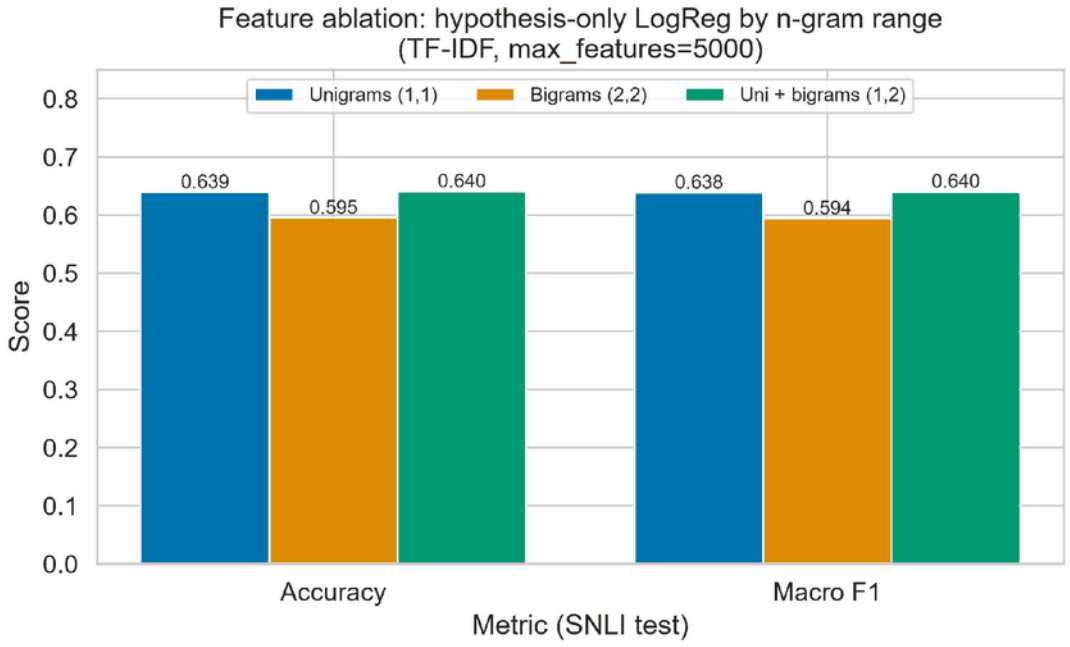


Figure 7. Accuracy and macro F1 of hypothesis-only logistic regression with unigram, bigram, or combined vocabularies.

5.4 Do the shortcuts transfer to MultiNLI?

Table 8 and Figure 8 depict the performance of the SNLI-trained models on MultiNLI when used without any fine-tuning. While all of the hypothesis-only models are still performing better than the baseline model, their improvement is significantly smaller. The hypothesis-only models outperform the baseline by 27–30 points on SNLI and 3.8–8.5 points on MultiNLI. The gain kept by hypothesis-only LR on the matched set and the mismatched set of MultiNLI is 24% and 26%, respectively; by hypothesis-only SVM — 25% and 28%; by hypothesis-only Naive Bayes —16% and 14%. The results obtained on the matched and mismatched sets of MultiNLI are quite similar, thus, the SNLI cues do not transfer better to some MultiNLI genres than to others.

Table 8. Accuracy of SNLI-trained models on SNLI test and on the MultiNLI validation sets, without retraining. "Gain kept" is the improvement over the dummy baseline on MultiNLI divided by the improvement on SNLI.

Model	SNLI test	MNLI matched	MNLI mismatched	Gain kept (matched)	Gain kept (mismatched)
Dummy (most frequent)	0.3428	0.3545	0.3522	—	—
LogReg (hyp-only)	0.6403	0.4247	0.4292	24%	26%
MultinomialNB (hyp-only)	0.6142	0.3981	0.3904	16%	14%
LinearSVC (hyp-only)	0.6406	0.4281	0.4369	25%	28%
LogReg (full input)	0.7005	0.4702	0.4738	32%	34%

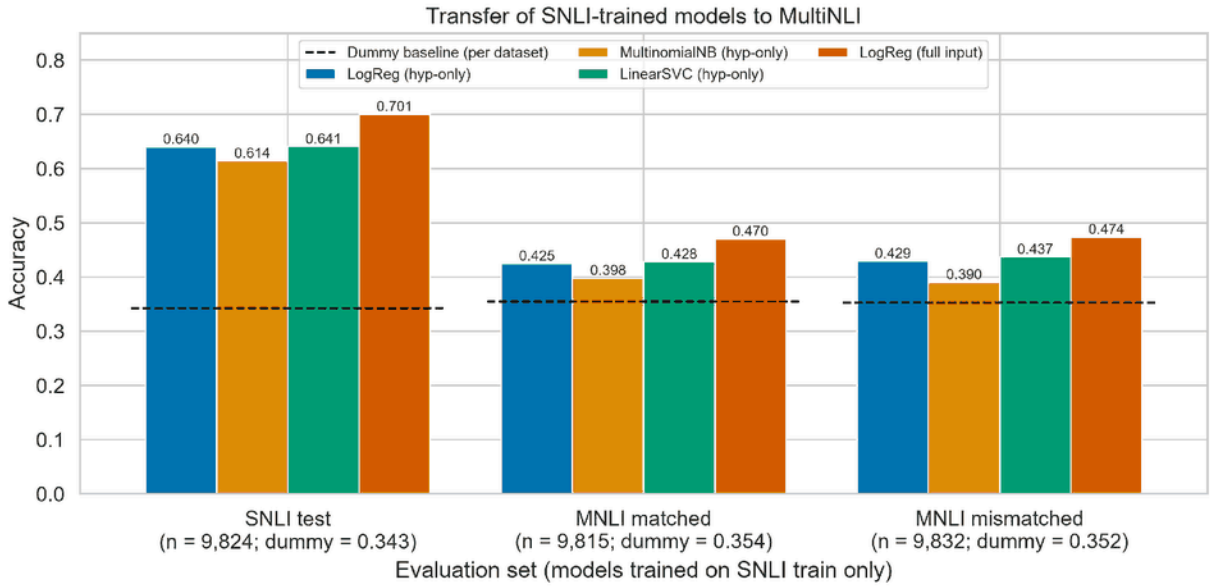


Figure 8. Accuracy of SNLI-trained models on SNLI and on MultiNLI (matched and mismatched). Dashed lines mark each dataset's majority baseline.

This drop requires a nuanced explanation. It suggests that quite a few SNLI shortcuts depend on how exactly SNLI is constructed, thus corroborating the hypothesis that they are artifacts of its annotation procedure. Nevertheless, the full-input model shows a clear drop as well, from 70.1% to 47.0% and 47.4%, thus losing about two-thirds of its advantage. The first explanation that comes to mind is the simple domain shift: MultiNLI consists of sentences from various genres, using vocabulary unknown to SNLI-trained feature extractors. The transfer test alone is not enough to differentiate between these two processes. The point made by the analysis is that a model, if it relies on SNLI's hypothesis hints to perform well, will be much less successful when applied to text in new domains. Gururangan et al. [4] found that hypothesis-only models on MultiNLI achieve around 53% accuracy, thus showing that MultiNLI contains its own artifacts, although they are certainly different from those in SNLI.

5.5 Does the premise-aware model actually use the premise?

It gives the clearest result in the easy/hard partition (Table 9 and Figure 9): the hypothesis-only logistic regression (LR) classifies 6,290 examples correctly (the easy set, 64.0%) and misclassifies 3,534 (the hard set, 36.0%). For the easy subset, the full-input LR has an accuracy of 91.6%, while in the hard subset, it reaches just 31.7%, showing the drop of 59.9 percentage points. Such accuracy on the hard subset is lower than the random guessing level; e.g., if one predicts the contradiction label (the most common in this set, 35.9%), the results will be better.

Table 9. Easy/hard split of the SNLI test set according to correctness of labeling by the hypothesis-only LR, along with the label distribution in each subset (entailment / neutral / contradiction, %) and the accuracy of the full-input LR in each subset.

Subset	Examples	% of test	Label mix (E / N / C)	Full-input accuracy	Full-input macro F1
easy	6,290	64.0%	35.9 / 32.8 / 31.3	0.9159	0.9157
hard	3,534	36.0%	31.4 / 32.7 / 35.9	0.3172	0.3148
all	9,824	100.0%	34.3 / 32.8 / 32.9	0.7005	0.6993

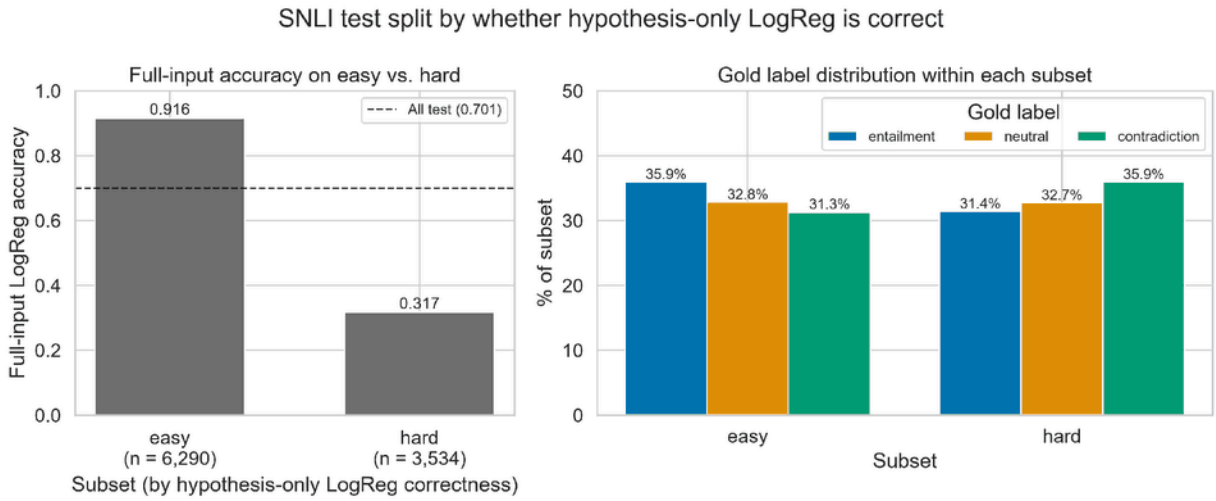


Figure 9. Left: accuracy of the full-input logistic regression on easy and hard examples compared to its general accuracy (dashed line). Right: label distribution inside each of the subsets. The labels of the hard set are close to balanced, so the difference is not caused by one of the labels overwhelming the set.

As the distribution of labels in the two subsets is fairly balanced (the hard set has only slightly more contradictions), the reason for the difference is probably not one of the labels dominating the other. The most plausible explanation is that the full-input classifier has learned similar cues of the hypothesis as the hypothesis-only model and uses them to make decisions successfully, failing only when the cues go in the wrong direction. There is, however, an important limitation to keep in mind: the full-input classifier also contains the hypothesis TF-IDF block, so it has some part of its input shared with the second classifier, which makes the division non-independent. The conclusion to be drawn is that, for bag-of-words classifiers, including the premise mostly improves performance on already easy examples and does not reflect the hardness of the hard examples for humans or for a stronger classifier.

5.6 Which words do the models rely on?

Table 10 lists the ten highest-ranked terms for each label using the three approaches to per-label explanations, and Figure 10 illustrates the ranking results using the SHAP method in more detail. Initially considering the rankings based on the coefficients and those from Naive Bayes reveals that there are three themes that match the annotation strategies used in past research [4]:

- **Contradiction**: negations and impossible scenarios. Terms related to negations appear as dominating both rankings, where "nobody" has the largest coefficient among all terms for all labels (6.65), followed by "nothing" (5.42) and "no" (5.10). In addition, there are "never" and "nobody is" terms. All the other terms refer to conditions under which it is easy to make a caption inconsistent with a certain image, such as "sleeping," "at home," "cats," "naked," "frowning," and even "mars" (in case of "on Mars").
- **Entailment**: vagueness and generalizations. The top terms help make a description more vague, like "something," "an animal," "humans," "at least," "some humans," "wearing

clothing," "are outdoors," and "interacting." They allow making a sentence true for nearly every image.

- **Neutral**: additional, not verifiable information. Terms add information not contained in an image, including emotions ("sad," "joyously," "happily," "funny"), physical qualities ("tall"), relations ("siblings," "brothers," "friends"), events ("championship," "vacation"), and causal explanations ("because," "for").

Table 10. The ten highest-ranked hypothesis terms per label for each explanation method (bigrams shown as two words). Full top-20 lists with scores are in Appendix B.

Label	Method	Top 10 terms
entailment	LR coefficient	making music, is near, something, not standing, interacting, this picture, near each, are by, are near, an animal
entailment	NB log-ratio	some humans, least two, picture and, this picture, than one, at least, least, wearing clothing, are outdoors, holding something
entailment	LR SHAP	people, person, there, in the, outside, there is, is in, are outside, is outside, on the
neutral	LR coefficient	joyously, sad, tall human, championship, siblings, sad man, favorite, tall, tall person, funny
neutral	NB log-ratio	tall human, tall humans, joyously, are siblings, tall people, are brothers, are sisters, tall person, are friends, sad man
neutral	LR SHAP	the, is, are, to, in, for, on, at, his, two
contradiction	LR coefficient	nobody, nothing, no, cats, frowning, never, at home, cat, mars, nobody is
contradiction	NB log-ratio	nobody is, nobody, nobody has, home sleeping, two cats, on mars, cats are, home watching, inside watching, no people
contradiction	LR SHAP	the, in, nobody, sleeping, at, woman, no, group of, nobody is, his

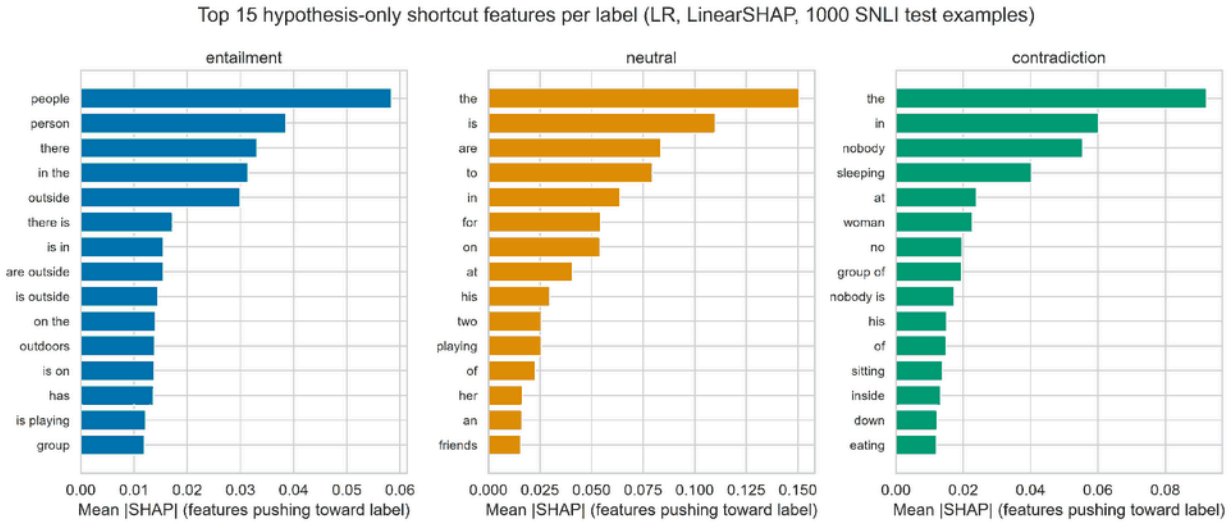


Figure 10 shows the fifteen highest-ranked hypothesis features per label based on LR SHAP importance (mean absolute SHAP value across 1,000 test examples, limited to those features which have an average effect favoring the particular label). In contrast to the coefficients and the Naive Bayes ranking, the most common words like "the", "is", and "in" obtain the highest ranks.

The SHAP ranking (Figure 10) tells a different story altogether. For the neutral label, the highest-ranked words are "the", "is", "are", "to", and "in"; for the contradiction label, the most prominent words include "the" and "in", followed by "nobody" and "sleeping"; while for entailment, the most prominent words are "people", "person", "there", "in the", and "outside". While some content words reappear here, they are interspersed among frequent function words. Figure 11 shows in more detail for the contradiction label that infrequent words like "nobody" and "no" can be associated with very large SHAP values (up to 6.6 on the log-odds scale for "nobody") despite their low frequency (appearing only in a small fraction of sentences), but frequent words like "the" receive small contributions each time but occur in about 43% of the hypotheses. Figure 12 shows that the three highest-ranked SHAP words ("the", "is", "are") have a tendency towards the neutral label when they occur, even though each contribution is individually small.

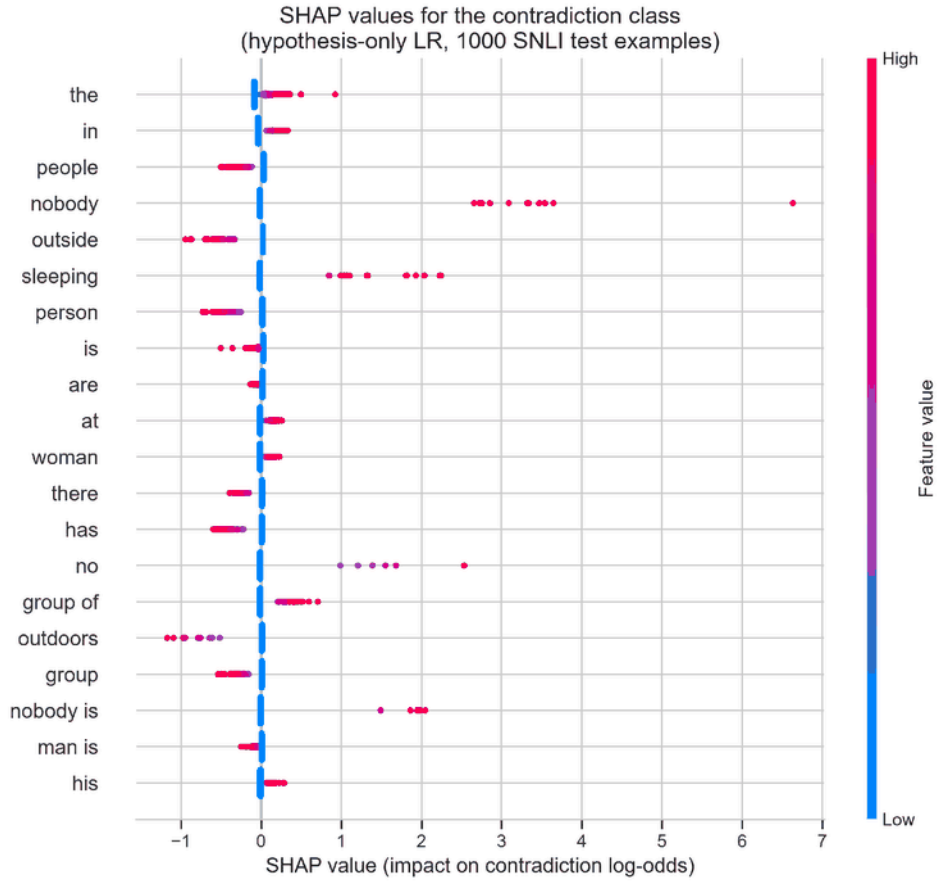


Figure 11. Contradiction label SHAP values have been calculated for the 20 features having the highest mean absolute SHAP values using a test dataset containing 1,000 samples. Every data point corresponds to an individual sample. The colors in the plot denote the TF-IDF score for that particular feature (where red indicates higher values and blue indicates lower values or absence). Negations like “nobody” and “no” are shown to have significant impact on the presence of the feature, while “outdoors,” “outside,” and “people” have significant impact on the opposite of contradiction.

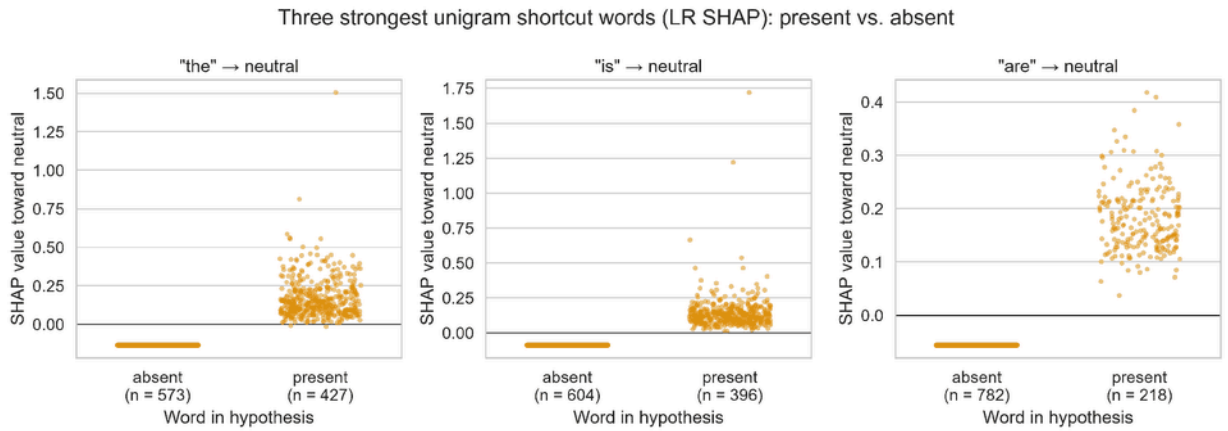


Figure 12 illustrates SHAP values for the neutral label corresponding to the three most important unigrams according to SHAP, conditional on the presence/absence of the word in the hypothesis, using 1,000 test instances. Each word consistently pulls toward neutral if it is present and marginally pulls away from neutral if it is not.

The question whether the words "the," "is," and "are" could be considered shortcuts is justified and discussed in Sections 5.7 and 5.8. Indeed, these words carry information in that their presence pulls the logistic regression (LR) predictions toward neutral, possibly because neutral hypotheses mention particular concrete people, places, and events, while entailed hypotheses use more general language, referring to e.g., "a person is outside." This interpretation is speculation and not tested empirically in this paper. However, these words do not serve as clues that readers would use as an annotator's technique.

5.7 Do the explanation methods agree?

Agreement is evaluated directly in Table 11 and Figure 13. The results show disagreement that is larger than could be expected. LR coefficients and Naive Bayes (NB) ratios have moderate agreement (Spearman ρ of 0.15−0.45 depending on the label, mean ρ = 0.33). There is almost no agreement between LR coefficients and LR SHAP (mean ρ = 0.06), although they are extracted from one and the same fitted model. LR SHAP and NB rankings disagree for all the labels (ρ between −0.71 and −0.47, mean −0.58). Random-forest Gini importance agrees strongly with LR SHAP (ρ = 0.89) but disagrees with NB (ρ = −0.75) and with LR coefficients (ρ = −0.27).

Table 11. Spearman rank correlation between explanation methods, computed per label over the union of the methods' top-100 terms (219−237 terms per label). The mean is reported across the three labels.

Method pair	Entailment	Neutral	Contradiction	Mean ρ
LR coef vs LR SHAP	0.23 (p < 0.001)	-0.07 (p = 0.27)	0.01 (p = 0.91)	0.06
LR coef vs NB log-ratio	0.15 (p = 0.03)	0.45 (p < 0.001)	0.40 (p < 0.001)	0.33
LR SHAP vs NB log-ratio	-0.55 (p < 0.001)	-0.71 (p < 0.001)	-0.47 (p < 0.001)	-0.58

Table 12. Agreement of random-forest Gini importance (one score per term) with each other method's average importance magnitude across labels, over the union of all four top-100 lists (276 terms).

Method pair	Spearman ρ
RF Gini vs LR coef	-0.27 (p < 0.001)
RF Gini vs LR SHAP	0.89 (p < 0.001)
RF Gini vs NB log-ratio	-0.75 (p < 0.001)

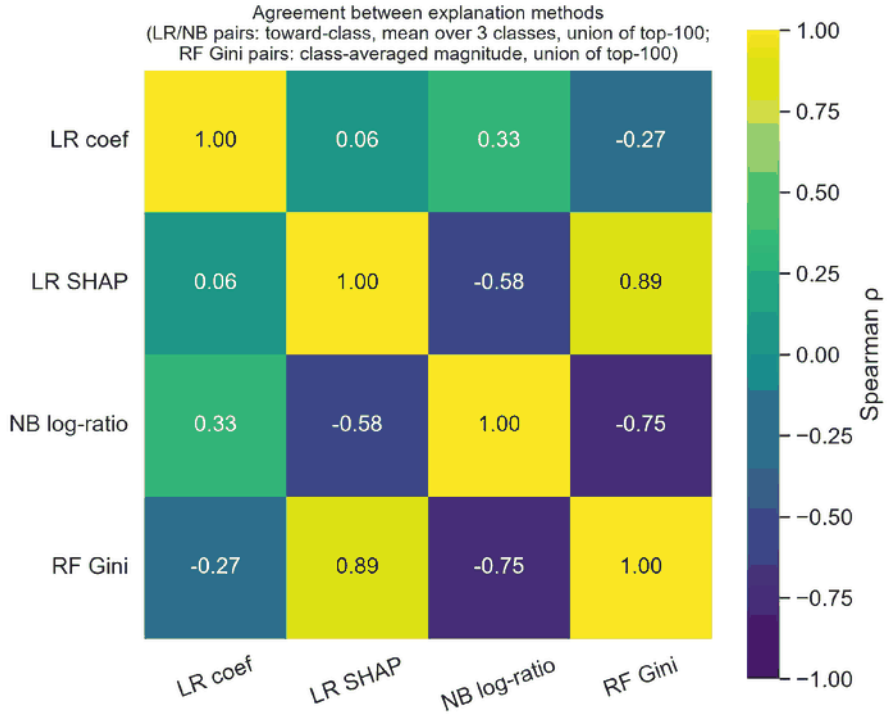


Figure 13 depicts the correlation among the four explanation approaches (Spearman ρ, averaged across labels for the per-label approaches). There are two clusters in the correlation matrix: LR SHAP and RF Gini are correlated with each other, LR coefficients and Naive Bayes approaches are moderately correlated with each other, but the clusters are not.

The correlation seen in Figure 13 can be explained by the term frequency analysis in Table 13. LR SHAP values correlate very strongly with the frequency of the term in training hypotheses (ρ = +0.86 to +0.90). In turn, Naive Bayes ratios correlate negatively with the frequency of the term (ρ = −0.57 to −0.78), and LR coefficients have very weak and negative correlations (ρ = −0.11 to −0.34). In other words, SHAP and Gini importance are frequency-based: the more often the term impacts the predictions, the higher importance score it receives. On the contrary, coefficients and Naive Bayes ratios are occurrence-based metrics: the higher the informativeness of each occurrence of the term, the higher importance score it will get, regardless of how frequent the term is. Naive Bayes favors rare terms, as the ratio of

probabilities becomes extremely high for a term that occurs dozens of times in the same label every time. Thus, for the same set of terms, the opposite behaviors result in negative correlations in Table 11.

Table 13. Spearman correlation between each method's per-label scores and the terms' training document frequency, over the same term sets as Table 11. All values have $p < 0.001$ except where shown.

Method	Entailment	Neutral	Contradiction
LR coef	-0.11 (p = 0.09)	-0.27	-0.34
LR SHAP	+0.86	+0.90	+0.86
NB log-ratio	-0.57	-0.78	-0.61

One of the hypotheses considered at an earlier stage of the results' analysis was that such function words as "the" could be used as proxies for hypothesis length because neutral hypotheses are generally longer (see Section 3.3). But there is no evidence supporting this hypothesis in the length-based analysis shown in Table 14 below. The per-example neutral SHAP values of "the," "is," and "are" have only small correlations $\rho = +0.08$, -0.07, and -0.06 with the length of hypotheses, respectively. There is a very small amount of prepositions correlated positively, although not strongly (for instance, "in" $\rho = +0.27$, "to" $\rho = +0.23$). Hence, the function words carry some label information on their own, and the high ranking of these SHAP features is due to their high frequency.

Table 14. Correlation between the ten highest-ranked LR SHAP features' per-example SHAP values for the neutral label and hypothesis length (in words) across 1,000 test examples. "Present" means the count of the hypotheses with the given feature among the sample.

Rank	Feature	Strongest toward	Present	ρ with length
1	the	neutral	427	+0.08
2	is	neutral	396	-0.07
3	are	neutral	218	-0.06
4	to	neutral	93	+0.23
5	in	neutral	190	+0.27
6	people	entailment	114	+0.03
7	nobody	contradiction	12	+0.17
8	for	neutral	33	+0.13
9	on	neutral	117	+0.17
10	at	neutral	78	+0.14

These results echo the discrepancy issue reported by Krishna et al. [12], and pinpoint its source in this particular setting: the applied methodologies do not give rise to contradictory responses to the same question, but ask different questions. The frequency-weighted methodology asks

what terms have the strongest cumulative effect on the predictions of the model. On the other hand, the per-occurrence methodology asks which terms have the highest diagnostic value when they occur. Both methodologies are valid; however, a researcher looking for annotation artifacts, i.e., special words that a subset of annotators use, will get very different results depending on which method is used.

5.8 Do the shortcut words actually matter?

The removal experiment compares the two ranking methodologies with the reality of the situation. Tables 15 and 16 and Figure 14 show the hypothesis-only logistic regression (LR) accuracy after removing the top k words selected by each ranking from the test hypotheses, together with a frequency-controlled random baseline. Without any words being removed, the accuracy is 64.03%.

Table 15. Removal experiment, LR SHAP ranking. Accuracy of the hypothesis-only LR after removing the top k words from test hypotheses, compared with the nearest-frequency-rank random control (mean ± SD over five seeds). “Extra drop” is the difference between the accuracy of the control and that of the shortcut in percentage points. Frequency is the average training document frequency of the removed words; “Changed” is the number of test hypotheses modified (both shortcut and control). “Sets” is the number of random word sets generated from the five seeds.

k	Shortcut removed	Random control	Extra drop (pts)	Frequency	Changed	Sets
5	0.6077	0.6424 ± 0.0000	3.47	147,464 / 18,007	7,939 / 1,522	1
10	0.5886	0.6396 ± 0.0000	5.10	93,477 / 12,756	8,583 / 2,088	1
20	0.5720	0.6361 ± 0.0002	6.41	61,157 / 9,571	9,151 / 3,072	4
50	0.5420	0.6246 ± 0.0004	8.26	30,961 / 6,591	9,470 / 4,529	5
100	0.5148	0.6157 ± 0.0008	10.09	19,219 / 4,735	9,687 / 5,746	5

Table 16. Removal experiment, LR coefficient ranking (words in at least 0.1% of training hypotheses). Columns as in Table 15.

k	Shortcut removed	Random control	Extra drop (pts)	Frequency	Changed	Sets
5	0.6320	0.6397 ± 0.0001	0.77	1,807 / 1,812	148 / 153	4
10	0.6156	0.6373 ± 0.0003	2.17	25,723 / 14,038	4,263 / 2,388	5
20	0.5927	0.6329 ± 0.0011	4.02	15,334 / 9,481	4,760 / 2,996	5
50	0.5609	0.6243 ± 0.0031	6.34	9,886 / 7,276	6,170 / 5,158	5
100	0.5221	0.6135 ± 0.0016	9.14	9,138 / 6,547	8,167 / 7,029	5

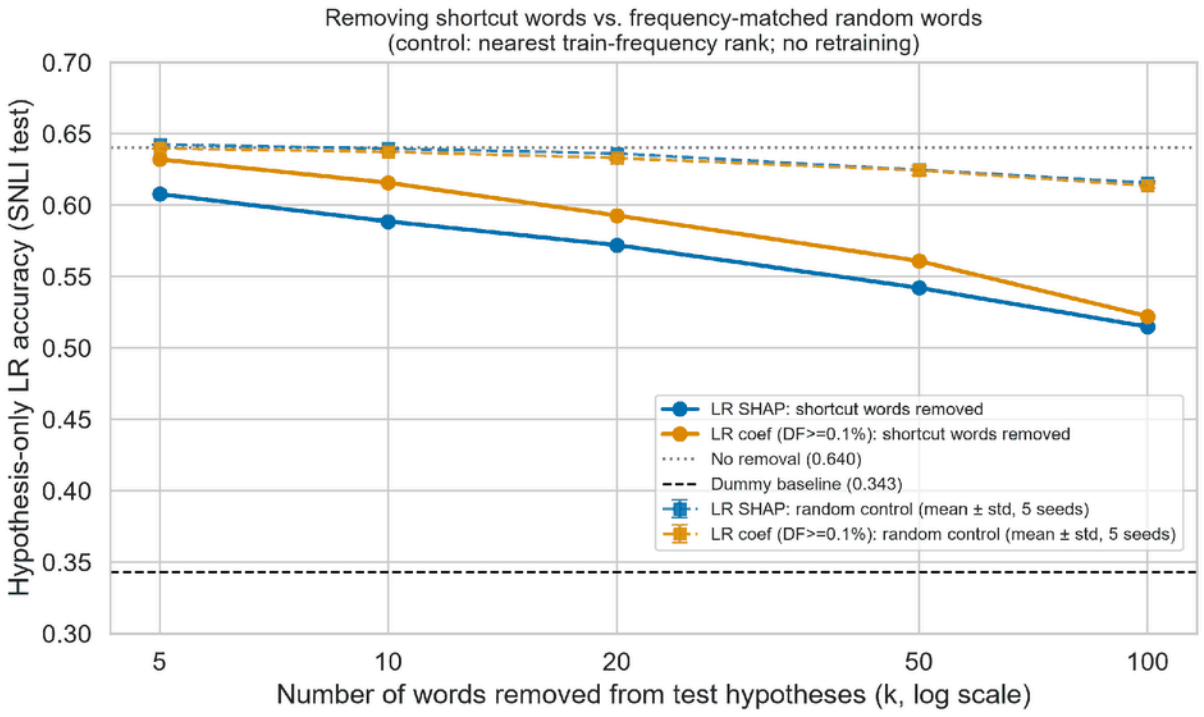


Figure 14. Hypothesis-only LR performance after removing the top k words according to the ranking (solid lines) or frequency-matched random words (mean ± SD over five seeds, dashed lines), without retraining. The dotted line shows accuracy with no removal, and the dashed black line is the dummy baseline. Both rankings outperform their respective random controls for all k. Removal of the top five words according to the SHAP ranking (the, is, are, to, in) leads to an accuracy of 60.77%, which is 3.47 points lower than the control accuracy; at k = 100, the additional decrease is 10.09 points. Removal of the top five according to the coefficient ranking (nobody, nothing, no, because, sad) yields an accuracy of 63.20%, which is 0.77 points lower than the control accuracy; at k = 100, the additional decrease is 9.14 points, with an accuracy of 52.21%.

The match quality dictates the interpretability of the comparison. At k = 5 in the coefficient ranking, the match is nearly perfect: the average frequency of the removed shortcut words (1,807) and that of the control words (1,812) are nearly equal, and both alter a nearly identical number of test hypotheses (148 vs 153). This is the clearest test in the study, indicating a real effect (albeit a small one): removal of five specific words from fewer than 2% of test hypotheses leads to a 0.77 point increase in additional loss compared to the removal of five comparable words. Since the five words alter 148 sentences, the effect is concentrated: compared to the no removal, there is a drop in accuracy of 0.83 points, equivalent to a net loss of about 82 correct predictions out of 148. At k = 50 and k = 100, the match is decent (frequency within 1.4×, altered hypotheses within 1.2×), and the additional losses of 6.34 and 9.14 points respectively indicate that collectively, the coefficient-ranked words are significantly more influential than random words with similar frequencies. At k = 10 and k = 20, the match becomes worse, partly due to “the” being in the top ten according to the coefficient ranking.

The contrast of the SHAP ranking is particularly difficult to interpret when k is small, since it includes among its top words "the", "is", and "are", which happen to be some of the most frequent words in the English language. In fact, there is no word of equal frequency that is not included in the SHAP ranking top 100 and hence not included in the control group. This is why when k is 5, the shortcut words are found in nearly eight times as many training hypotheses as the control group (147,464 vs 18,007 on average), and change nearly five times as many test hypotheses (7,939 vs 1,522). The greater decrease in accuracy of the SHAP ranking is therefore hard to disentangle from the fact that it simply deletes much more text. This is a flaw of our experiment design and not evidence against SHAP, but it means that the best evidence we have for causality supports the words selected via the coefficients.

There are two further details worth pointing out. First, since the redesign always picks the closest word by frequency rank, the results on five seeds often coincide for each k. For the SHAP ranking, we obtained just one unique set for k = 5 and k = 10, respectively. Hence, the very low standard deviations in the control column do not indicate anything unusual but merely reflect the near-deterministic choice procedure, and should not be used for computing "number of standard deviations". Second, the original bin-matched control design (Appendix C) yields qualitatively identical results, where both rankings beat their controls for all k.

5.9 Error analysis

Table 17 shows the 15 randomly sampled examples, which make the aggregate patterns concrete.

Table 17. Randomly sampled test examples (seed 42). Group A: hypothesis-only LR correct with confidence > 0.9; B: wrong with confidence > 0.9; C: hypothesis-only wrong but full-input correct. Labels: E = entailment, N = neutral, C = contradiction; confidence in parentheses. The last column lists the three present features with the highest SHAP values toward the hypothesis-only prediction. Premises are in the project repository.

Group	Hypothesis	Gold	Hyp-only	Full input	Top SHAP features
A	A woman is sleeping.	C	C (0.94)	C (0.94)	sleeping +1.931; is sleeping +0.528; woman +0.171
A	A girl sleeping on the ground.	C	C (0.91)	C (0.97)	sleeping +1.292; ground +0.287; sleeping on +0.124
A	The man is alone sleeping in his bedroom on the moon.	C	C (0.99)	C (1.00)	sleeping +0.747; alone +0.745; the moon +0.735
A	A girl captures a butterfly for a science fair project	N	N (0.92)	N (0.96)	for +0.851; fair +0.834; project +0.682
A	The boy is playing video games upstairs in his room.	C	C (0.95)	C (0.96)	video games +0.687; playing video +0.672; his room +0.540

Group	Hypothesis	Gold	Hyp-only	Full input	Top SHAP features
B	A group of kids are outdoors.	N	E (0.91)	E (0.95)	are outdoors +1.011; outdoors +0.588; group +0.208
B	A runner in a competition want to go to the finish line.	E	N (0.90)	N (0.87)	competition +0.646; to +0.587; want +0.301
B	2 male humans are working construction.	N	E (0.94)	E (0.79)	humans +0.932; are working +0.445; humans are +0.424
B	A professional bowling team wins the championship.	C	N (0.97)	N (0.98)	championship +1.897; professional +1.167; the championship +0.468
B	A black dog found a ball in the water and is bring it back to its owner.	E	N (0.92)	N (0.94)	owner +0.782; found +0.369; to +0.294
C	The stage crew is cleaning the stage	C	E (0.41)	C (0.62)	is cleaning +0.200; the stage +0.147; crew is +0.062
C	A bald man is playing a guitar.	C	E (0.45)	C (0.44)	is playing +0.350; bald man +0.065; playing guitar +0.023
C	A boy throwing a stone.	E	C (0.51)	E (0.78)	throwing +0.865; boy -0.079; stone -0.285
C	A girl might fall off a tree.	N	C (0.48)	N (0.60)	tree +0.356; off +0.211; girl -0.001
C	The children are watching the man blow up balloons	E	C (0.46)	E (0.51)	blow +0.300; watching +0.282; the +0.168

Group A shows the shortcuts working. "A woman is sleeping" is labeled contradiction with 94% confidence from the hypothesis alone, driven almost entirely by "sleeping" (SHAP +1.93); the premise, about hitting a tennis ball, is never consulted. "The man is alone sleeping in his bedroom on the moon" combines three contradiction cues, and "A girl captures a butterfly for a science fair project" is correctly called neutral because of the purpose clause "for ... project".

Group B shows the same cues failing. "A group of kids are outdoors" is gold neutral, but "are outdoors" is a strong entailment cue, so the model predicts entailment with 91% confidence. "A professional bowling team wins the championship" is gold contradiction (the premise describes a child at a bowling alley), but "championship" and "professional" are strong neutral cues. Strikingly, in all five group B examples the full-input model makes the same wrong prediction, in four of them with confidence above 0.85. Seeing the premise did not help it notice that a child is not a professional team.

Group C shows where the premise does help. In each case the hypothesis-only model was uncertain (confidence 0.41 to 0.51), with no strong cue present, and the full-input model used the premise to tip the decision. In "A bald man is playing a guitar" versus a premise about a blond man, for example, the correct contradiction depends on one word in each sentence. These are exactly the examples that require comparing the two sentences, and they make up only part of the test set.

6. Discussion

6.1 How much of NLI performance is a shortcut?

With regards to the first research question, the answer is clear. Without having access to the premise, standard linear classifiers trained on SNLI are able to classify 64% of the test examples, achieving an accuracy rate that is almost twice as high as the expected rate of 34% if we take into account pure randomness, and they explain 83% of the improvement over chance that the same features can achieve if the premise is provided. The cues used to make such classifications come from a mere thousand examples, are mostly contained in single words, and the premises-informed classifier succeeds precisely on those cases where the cues lead to the right answer. This set of results suggests that, at least for bag-of-words models on SNLI, most of the performance achieved is independent of the comparison between the two sentences.

This does not mean that more advanced models, such as large pretrained transformers, do not use shortcuts to that extent; they can reach an accuracy of around 90% on SNLI and use the premise in their predictions. However, it shows that a certain percentage of each classifier's success on SNLI can be obtained without reasoning, and the benchmark in itself is not enough to differentiate models that reason about the input from those that have internalized the conventions of the annotators. Data created with the purpose of breaking particular shortcuts, like HANS, as well as adversarial data, still have their place next to test sets.

6.2 Explanation methods answer different questions

The answer to the second research question was the most surprising one. Four valid explanation approaches, two of which came from the same fitted model, provided strikingly different ranks of the words' importance with two of the method pairs almost reversing the stated order. An analysis of frequencies suggests that this difference does not come from randomness since all the methods are systemically different in how they account for frequency. Namely, the average SHAP importance value and random forest Gini importance tend to prioritize the words that affect a large number of predictions, thus preferring very frequent function words. On the other hand, logistic regression coefficients and Naive Bayes ratio give more weight to the words that are highly discriminative on a per-occurrence basis and thus prefer distinctive, rare words, where Naive Bayes seems to have a stronger preference for rare words.

Finally, the removal experiment brings an additional important observation: the per-occurrence terms that are selected by the coefficients ("nobody," "nothing," and "no") are causally effective in reducing predictions even after accounting for frequency. SHAP-ranked function words also decrease prediction accuracy upon their removal; however, since there is no proper comparison group of control words, the effect is confounded by removing larger pieces of text. Therefore, for the particular task of annotation artifact discovery, per-occurrence metrics seem to be more

useful while frequency-weighted importance should be considered with this particular type of bias in mind. As a general rule of thumb for those working with explanations, we advise explicitly reporting both the used explanation approach and its measure, comparing at least two approaches, and validating crucial observations via the controlled experiment like removal (see [12], [22]).

6.3 Connection to neuron-level explanations

The words selected in this study match very closely with those words found via neuron-level explanations for neural NLI systems [14], which include negative words like "nobody," as well as words such as "sleeping" and "eating" which occur in the hypothesis. This study presents a simple explanation for the emergence of such neurons: each of these single words contains a lot of label information in SNLI, and thus can be learned by a simple linear model, which has no access to the premise, based on a few thousand examples. It makes sense for a neural network trained to reduce errors on this data set to have an interest in learning these words [7]. The fact that NLI neurons that can be described using simple rules tend to have lower predictive accuracy [14] follows quite well from this explanation, since the rules formed from these words are successful on many easier examples but not on harder examples, as seen from the easy/hard analysis in Section 5.5 of bag-of-words systems.

6.4 Limitations

There are several limitations to this work that affect the interpretation of its findings.

- **Simplistic models and features.** All the models used only bag-of-words TF-IDF features with vocabulary size of 5,000 and no hyperparameters tuning. Better models, a larger vocabulary size, or word embeddings would most probably yield better results (Gururangan et al. report 67.0% accuracy with fastText [4]). The results show what a simplistic model learns; not what any model learns.
- **Only one training dataset**. The shortcuts were learned only on SNLI dataset. While the MultiNLI experiment is done to examine the model transferability, it cannot distinguish between a specific shortcut-related problem and a more general domain shift effect, as the full-input model shows an equally sharp decrease in performance.
- **Shared features in the easy-hard split**. There are hypothesis-specific features inside the full-input model, so a part of the failures of the latter on hard examples are due to those very features. The split shows that adding premise makes a difference for easy examples only; it does not measure how hard the examples are in absolute terms.
- **Inefficient removal controls.** In case of the SHAP ranking, there is no equivalent control words by frequency, so SHAP removal experiments at small k are confounded by the proportion of text removed. The experimental design was changed after preliminary experiments because of poor matching. Both results are presented, and they both agree in direction.
- **Explanation coverage issues.** The random forest SHAP values could not be computed on the current computer, hence SHAP of the forest is only Gini importance and provides a single score per word, not per label. Logistic regression SHAP values were computed on a sample of 1,000 test examples.

- **No retraining after removal.** Removal of the words from test sentences allows to examine whether the model uses them in the process; however, it does not show what will happen if the word is removed from training and not what would the model learn if the words are removed.

6.5 Future work

A number of possibilities for extension arise readily. The same pipeline could be used for transformer models which have been pretrained to investigate whether these models utilize the same lexical features and how frequently they use different explanation methods. Hypothesis-only models can be trained using MultiNLI and other natural language inference (NLI) corpora to allow for comparisons between artifacts seen in this study and artifacts that could arise in SNLI. The removal test can be performed again with shortcut words being eliminated from the training dataset to see if the models now learn to rely on the premise. Lastly, merging the data-driven approach of the current paper with neuron explanations for the same model would allow to check if the neurons sensitive to the words like “nobody” drive the shortcut behavior of the model.

7. Conclusion

This paper explores what can a model figure out about natural language inference without the premise. In the context of the SNLI dataset, the results are impressive. Basic classifiers using only the hypothesis achieve 64% accuracy, in comparison with 34% in case of random guessing and 70% with the similar model analyzing both sentences. The model picks up hints from a couple of thousands examples, and these hints consist mostly of single words and correspond to the strategy used by the annotator when building data: contradictions are indicated by negations and incompatible actions, entailments are recognized due to vague generalizations, and neutrals are based on additional information. The model doing the analysis with premises performs well when these hints work and underperforms in comparison with random guessing otherwise; additionally, these hints partially transfer to another dataset.

This study shows that the explanation of such models' shortcut is not as simple as using any explanation method. Four standard approaches have large disagreements on what words are significant because they measure word frequency in opposite ways, and only the proper removal experiment can provide the list of words used by the model. The message from this work is that both benchmark accuracy and the explanation should be tested in controlled experiments before being considered reliable. Shortcuts are easy to learn by a machine, while the tools to detect them are solving different problems.

Acknowledgments

This independent project was inspired by my earlier internship experience in neural explanation research. The research questions, the experimental design decisions, the execution of all experiments and the review and interpretation of every result were carried out by me, and all reported numbers were produced by the released code.

Code and Data Availability

All code, result files and figures are available at https://github.com/sainellutla/premise-free-nli. The repository includes a log of every problem encountered and every design decision made during the project. SNLI and MultiNLI are publicly available through the HuggingFace Hub.

References

[1] I. Dagan, O. Glickman, and B. Magnini, "The PASCAL recognising textual entailment challenge," in *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, Lecture Notes in Computer Science, vol. 3944. Berlin, Germany: Springer, 2006, pp. 177–190, doi: 10.1007/11736790_9.

[2] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. 2015 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015, pp. 632–642, doi: 10.18653/v1/D15-1075.

[3] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, New Orleans, LA, USA, 2018, pp. 1112–1122, doi: 10.18653/v1/N18-1101.

[4] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, "Annotation artifacts in natural language inference data," in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Vol. 2 (Short Papers)*, New Orleans, LA, USA, 2018, pp. 107–112, doi: 10.18653/v1/N18-2017.

[5] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme, "Hypothesis only baselines in natural language inference," in *Proc. Seventh Joint Conf. Lexical and Computational Semantics (*SEM)*, New Orleans, LA, USA, 2018, pp. 180–191, doi: 10.18653/v1/S18-2023.

[6] M. Tsuchiya, "Performance impact caused by hidden bias of training data for recognizing textual entailment," in *Proc. Eleventh Int. Conf. Language Resources and Evaluation (LREC)*, Miyazaki, Japan, 2018. [Online]. Available: https://aclanthology.org/L18-1239/

[7] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020, doi: 10.1038/s42256-020-00257-z.

[8] T. McCoy, E. Pavlick, and T. Linzen, "Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference," in *Proc. 57th Annu. Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019, pp. 3428–3448, doi: 10.18653/v1/P19-1334.

[9] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela, "Adversarial NLI: A new benchmark for natural language understanding," in *Proc. 58th Annu. Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 4885–4901, doi: 10.18653/v1/2020.acl-main.441.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, San Francisco, CA, USA, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[11] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 4765–4774. [Online]. Available: https://arxiv.org/abs/1705.07874

[12] S. Krishna, T. Han, A. Gu, S. Wu, S. Jabbari, and H. Lakkaraju, "The disagreement problem in explainable machine learning: A practitioner's perspective," arXiv preprint arXiv:2202.01602, 2022. [Online]. Available: https://arxiv.org/abs/2202.01602

[13] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, Montréal, Canada, 2018, pp. 9505–9515. [Online]. Available: https://arxiv.org/abs/1810.03292

[14] J. Mu and J. Andreas, "Compositional explanations of neurons," in *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020, pp. 17153–17163. [Online]. Available: https://arxiv.org/abs/2006.14032

[15] D. Kaushik and Z. C. Lipton, "How much reading does reading comprehension require? A critical investigation of popular benchmarks," in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, 2018, pp. 5010–5015, doi: 10.18653/v1/D18-1546.

[16] T. Schuster, D. Shah, Y. J. S. Yeo, D. Roberto Filizzola Ortiz, E. Santus, and R. Barzilay, "Towards debiasing fact verification models," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3419–3425, doi: 10.18653/v1/D19-1341.

[17] Y. Belinkov, A. Poliak, S. Shieber, B. Van Durme, and A. Rush, "Don't take the premise for granted: Mitigating artifacts in natural language inference," in *Proc. 57th Annu. Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019, pp. 877–891, doi: 10.18653/v1/P19-1084.

[18] C. Clark, M. Yatskar, and L. Zettlemoyer, "Don't take the easy way out: Ensemble based methods for avoiding known dataset biases," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 4069–4082, doi: 10.18653/v1/D19-1418.

[19] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games, Vol. II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton, NJ, USA: Princeton Univ. Press, 1953, pp. 307–317, doi: 10.1515/9781400881970-018.

[20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[21] S. Jain and B. C. Wallace, "Attention is not explanation," in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 3543–3556, doi: 10.18653/v1/N19-1357.

[22] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in *Advances in Neural Information Processing Systems 32 (NeurIPS)*, Vancouver, Canada, 2019, pp. 9737–9748. [Online]. Available: https://arxiv.org/abs/1806.10758

[23] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6541–6549, doi: 10.1109/CVPR.2017.354.

[24] Q. Lhoest *et al.*, "Datasets: A community library for natural language processing," in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing: System Demonstrations*, Punta Cana, Dominican Republic, 2021, pp. 175–184, doi: 10.18653/v1/2021.emnlp-demo.21.

[25] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html

[26] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0.

[27] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. AAAI-98 Workshop on Learning for Text Categorization*, Madison, WI, USA, 1998, pp. 41–48. [Online]. Available: https://www.cs.tufts.edu/~nr/cs257/archive/andrew-mccallum/event-models.pdf

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[29] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904, doi: 10.2307/1412159.

Appendix A. Reproducibility Details

All experiments were run in a Python virtual environment on Windows 11 (CPU only, 16 GB RAM, 12 logical CPUs). The exact package versions are pinned in the repository file requirements-lock.txt; the main ones are listed in Table 18 together with the model settings. The scripts are run in the order data.py, evaluate.py, rerun_rf_timing.py, learning_curve.py, feature_ablation.py, transfer.py, easy_hard.py, explain.py, removal.py and errors.py. The script evaluate.py saves the trained models and fitted vectorizers, which all later steps reuse, so that every analysis refers to exactly the same fitted models.

Table 18. Software versions and model settings. Settings not listed are scikit-learn 1.9.1 defaults.

Item	Setting
scikit-learn / shap / datasets	1.9.1 / 0.54.1 / 5.1.0
numpy / pandas / scipy	2.5.4 / 3.1.0 / 1.18.1

Item	Setting
matplotlib / seaborn	3.11.2 / 0.13.2
Random seed	42 everywhere (learning curve: 42, 43, 44; removal controls: 42 to 46)
TF-IDF	ngram_range = (1, 2), max_features = 5,000, lowercase; fit on training data only
Full-input features	separate premise and hypothesis TF-IDF + 3 overlap features (standardized on training data)
Dummy classifier	strategy = most_frequent
Logistic regression	lbfgs solver (multinomial), max_iter = 1,000
Multinomial Naive Bayes	alpha = 1.0
Linear SVM	LinearSVC defaults (C = 1.0)
Random forest	200 trees, fully grown, stratified 100,000-example training subsample
Cross-validation	5-fold stratified, shuffle, TF-IDF refit in each fold
SHAP	LinearExplainer (interventional), background = training-set feature means, 1,000 test examples

Appendix B. Full Explanation Rankings

Tables 19 to 21 list the 20 highest-ranked hypothesis terms per label for each per-label explanation method, with their scores.

Table 19. Top 20 terms per label by logistic regression coefficient (score in parentheses).

Rank	Entailment	Neutral	Contradiction
1	making music (2.82)	joyously (5.68)	nobody (6.65)
2	is near (2.77)	sad (4.23)	nothing (5.42)
3	something (2.67)	tall human (4.09)	no (5.10)
4	not standing (2.66)	championship (4.05)	cats (4.14)
5	interacting (2.61)	siblings (3.88)	frowning (3.85)
6	this picture (2.60)	sad man (3.82)	never (3.66)
7	near each (2.57)	favorite (3.65)	at home (3.64)
8	are by (2.51)	tall (3.54)	cat (3.64)
9	are near (2.45)	tall person (3.49)	mars (3.62)
10	an animal (2.41)	funny (3.47)	nobody is (3.57)
11	at least (2.38)	happily (3.46)	naked (3.54)
12	in this (2.37)	because (3.43)	sleeping (3.53)

Rank	Entailment	Neutral	Contradiction
13	least (2.36)	tall humans (3.43)	only (3.50)
14	wearing clothes (2.36)	famous (3.41)	jacks (3.47)
15	humans (2.36)	for (3.36)	silent (3.42)
16	motion (2.35)	brothers (3.23)	anything (3.41)
17	are outdoors (2.34)	friends (3.17)	quietly (3.26)
18	knows how (2.31)	vacation (3.17)	solid (3.19)
19	wearing clothing (2.30)	sisters (3.08)	don (3.11)
20	human (2.25)	owner (3.07)	one person (3.08)

Table 20. Top 20 terms per label by Naive Bayes log-probability ratio (label versus the other two labels).

Rank	Entailment	Neutral	Contradiction
1	some humans (4.63)	tall human (5.75)	nobody is (6.12)
2	least two (3.92)	tall humans (5.20)	nobody (6.08)
3	picture and (3.65)	joyously (4.97)	nobody has (6.03)
4	this picture (3.39)	are siblings (4.53)	home sleeping (4.86)
5	than one (3.38)	tall people (4.36)	two cats (4.74)
6	at least (3.30)	are brothers (4.27)	on mars (4.72)
7	least (3.24)	are sisters (4.24)	cats are (4.12)
8	wearing clothing (3.22)	tall person (4.20)	home watching (4.04)
9	are outdoors (2.96)	are friends (3.91)	inside watching (4.00)
10	holding something (2.91)	sad man (3.83)	no people (3.96)
11	people outdoors (2.88)	two sad (3.80)	mars (3.96)
12	is outdoors (2.84)	impress (3.75)	is naked (3.68)
13	near each (2.80)	to impress (3.73)	are no (3.68)
14	humans are (2.80)	for competition (3.65)	is sleep (3.60)
15	motion (2.72)	are married (3.65)	frowning (3.49)
16	are multiple (2.71)	sad woman (3.59)	his couch (3.43)
17	interacting (2.71)	championship (3.51)	no one (3.43)
18	is more (2.68)	practicing for (3.49)	are sleeping (3.40)
19	people present (2.67)	is winning (3.46)	nothing (3.37)
20	has something (2.65)	best friends (3.45)	is no (3.36)

Table 21. Top 20 terms per label by LR SHAP importance toward the label (mean absolute SHAP value over 1,000 test examples).

Rank	Entailment	Neutral	Contradiction
1	people (0.0584)	the (0.1505)	the (0.0922)
2	person (0.0385)	is (0.1098)	in (0.0602)
3	there (0.0331)	are (0.0835)	nobody (0.0554)
4	in the (0.0314)	to (0.0794)	sleeping (0.0401)
5	outside (0.0300)	in (0.0638)	at (0.0239)
6	there is (0.0172)	for (0.0543)	woman (0.0227)
7	is in (0.0155)	on (0.0540)	no (0.0196)
8	are outside (0.0155)	at (0.0408)	group of (0.0194)
9	is outside (0.0144)	his (0.0297)	nobody is (0.0172)
10	on the (0.0140)	two (0.0257)	his (0.0151)
11	outdoors (0.0139)	playing (0.0256)	of (0.0149)
12	is on (0.0138)	of (0.0227)	sitting (0.0138)
13	has (0.0136)	her (0.0164)	inside (0.0133)
14	is playing (0.0121)	an (0.0162)	down (0.0122)
15	group (0.0120)	friends (0.0156)	eating (0.0121)
16	child (0.0115)	man is (0.0142)	at home (0.0114)
17	at the (0.0114)	park (0.0135)	not (0.0102)
18	human (0.0112)	after (0.0131)	cat (0.0100)
19	something (0.0110)	new (0.0131)	on (0.0095)
20	are in (0.0109)	some (0.0124)	nothing (0.0088)

Appendix C. Original Removal Control

The first random control for the removal experiment grouped candidate words into 20 bins of equal size by training document frequency and replaced each shortcut word with a random word from the same bin. Table 22 and Figure 15 report its results. The frequency columns show the problem that motivated the redesign described in Section 4.7: for the SHAP ranking at k = 5, the shortcut words appear in about 22 times as many training hypotheses as the control words (147,464 vs 6,577). Despite this, the qualitative conclusion is the same as with the redesigned control: both rankings lower accuracy more than their controls at every k.

Table 22. Removal results with the original 20-bin frequency-matched control (mean ± SD over five seeds). Columns as in Table 15.

Ranking	k	Shortcut removed	Random control	Extra drop (pts)	Frequency	Changed
LR SHAP	5	0.6077	0.6390 ± 0.0019	3.13	147,464 / 6,577	7,939 / 563
LR SHAP	10	0.5886	0.6376 ± 0.0013	4.90	93,477 / 6,924	8,583 / 1,173
LR SHAP	20	0.5720	0.6340 ± 0.0011	6.20	61,157 / 7,155	9,151 / 2,292
LR SHAP	50	0.5420	0.6281 ± 0.0007	8.61	30,961 / 6,509	9,470 / 4,490
LR SHAP	100	0.5148	0.6175 ± 0.0020	10.27	19,219 / 4,551	9,687 / 5,602
LR coef	5	0.6320	0.6394 ± 0.0009	0.74	1,807 / 1,797	148 / 152
LR coef	10	0.6156	0.6380 ± 0.0025	2.24	25,723 / 3,479	4,263 / 622
LR coef	20	0.5927	0.6330 ± 0.0024	4.03	15,334 / 4,173	4,760 / 1,385
LR coef	50	0.5609	0.6273 ± 0.0029	6.64	9,886 / 4,660	6,170 / 3,496
LR coef	100	0.5221	0.6178 ± 0.0020	9.57	9,138 / 5,053	8,167 / 5,991

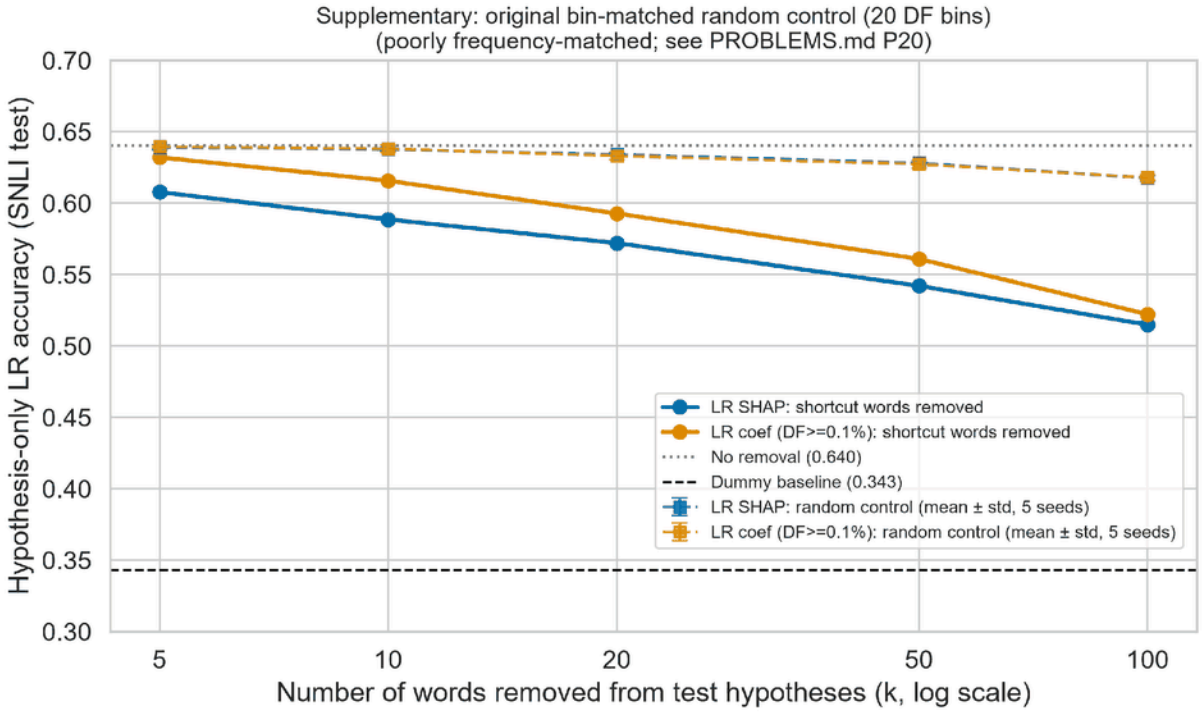


Figure 15. Removal results with the original bin-matched random control (supplementary). Solid lines remove shortcut words; dashed lines remove bin-matched random words.

Appendix D. Selected Problems and Design Decisions

The repository's PROBLEMS.md file records 24 problems and 26 design decisions. Table 23 summarizes those that affect how the results should be read.

Table 23. Selected problems and decisions from the project log.

Issue	What happened and how it was handled
Random forest timing	The laptop entered sleep mode during the first forest fit, inflating its recorded time to 7,865 s. The forest was refit with sleep blocked (277 s); accuracy and all 9,824 test predictions were identical.
Random forest SHAP	Exact tree SHAP did not finish one example in about eight minutes (fully grown trees average about 39,000 leaves), implying days for 1,000 examples. It was dropped; Gini importance is used instead.
SHAP memory stall	Passing the full training matrix as the SHAP background stalled the laptop. The exact training-set feature means were used instead, and SHAP values were verified to equal coefficient × (feature − mean) exactly.
Toward-label rankings	All per-label rankings order terms by how much they push toward the label, so that signed and unsigned scores are compared consistently in the agreement analysis.
Removal control redesign	The original 20-bin control was poorly frequency-matched for common words. It was replaced, after the first results were seen, by nearest-frequency-rank matching; both are reported (Section 4.7, Appendix C).
Near-deterministic control	The nearest-rank control often produced the same word set for all five seeds, so its standard deviations are near zero by construction.
Empty bigram inputs	With bigrams only, 509 test hypotheses had no vocabulary feature and were predicted from the model intercept alone.
Plot reproducibility	Two figures used unseeded random jitter; seeding was added and two runs were confirmed byte-identical.