Universiteit Utrecht

Mitigating Bias in the SNLI Dataset

July 16, 2021

by Afra Baas

Supervisor:
Lasha Abzianidze

Second Assessor:
Marijn Schraagen

Faculty of Humanities
BSc Artificial Intelligence
Credits: 7.5 ECTS

Contents

Mitigating Bias In The SNLI Dataset

Afra Baas (a.a.baas@students.uu.nl - 6517501)

July 2021, Utrecht University

Abstract

Natural language inference (NLI) is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise". An inference contains two sentences, where the first sentence is considered the premise and the second the hypothesis. NLI models have been performing very well. However, recent studies have shown the performance of NLI models trained on the Stanford Natural Language Inference (SNLI) dataset are overestimated. Hypothesis-only classifiers can already label most of the inferences correctly without looking at the premise. This suggests that NLI models depend heavily on annotation artifacts present in the hypothesis. A bias seems to occur because certain annotation artifacts are often in relation to a specific label.
This thesis will explore debiasing the SNLI dataset, which has this hypothesis-only bias. New examples will be generated to balance the annotation artifacts over the different labels. A few simple data augmentation techniques will be used to generate new premises in a way that the hypotheses will relate to 3 different labels, depending on the premise.
The results show that the hypothesis-only bias in the augmented SNLI dataset has decreased and that DistilBERT, one of the high-performing NLI models, maintains similar performance after fine-tuning on this augmented dataset. Balancing the annotation artifacts over all labels is therefore an efficient way to mitigate hypothesis-only bias in a dataset.

Keywords: Natural Language Inference; Annotation Artifacts; Data Bias; Data Augmentation; Hypothesis-only Bias

1 Introduction

Natural language inference Natural language processing (NLP) is a field in Artificial intelligence that concerns itself with automating tasks that require natural language understanding. These can be question answering, text summarization, Natural language inference, etc. In this thesis the focus will lie on a challenge in Natural language inference (NLI). Examples of inferences are given in table 1.

SNLI As NLI models use machine learning for inference classification task, and perform best on a large labeled dataset. Research in this field changed with the introduction of bigger labeled datasets. One of these large datasets was created by Bowman et al. (2015), whom created the SNLI dataset with the help of crowd workers. The SNLI corpus contains around 570k premise/hypothesis pairs and is manually labeled. The SNLI dataset was created by collecting captions (scene descriptions) from the Flickr30k corpus (Young et al., 2014) to serve as premises. The Flickr30k corpus is a collection of about 160k captions relating to 30k images, also collected using crowdsourcing.
The hypotheses in the SNLI dataset were obtained by presenting crowd workers with some of these premises and asking them to supply a hypothesis for each of the three labels: entailment, contradiction, neutral. For each inference in the train dataset the attributes 'annotator_labels' and 'gold_label' are the label for which the hypothesis was created. For the development and test datasets there was a validation phase, crowd workers were asked to give a label to some of the created inferences in these datasets. The attributes 'annotator_labels' of each inference in the development and test dataset consists of 5 labels, 4 labels obtained by validation and added to the attribute 'annotator_labels' with the label intended by the original author. Besides that, these inferences were also given a gold-label, which would be the label given to the inference three or more times. If there was no consensus, meaning any one of the labels was given less then three times, the gold-label would be the placeholder label '-'. Figure 1 also shows the 5 labels as the first letter of the label and the full label is the gold-label.
The SNLI dataset became a very popular dataset for training NLI models, but recently Gururangan et al. (2018); Poliak et al. (2018) showed that the SNLI dataset could use improvement.

Hypothesis-only bias Gururangan et al. (2018) showed that the text-classifier fastText[1] (Joulin et al., 2016) was able to classify 67% of the hypotheses correctly, while the majority baseline is 34%. Note that the majority baseline always predicts the majority class, and is important for reference when using the accuracy metric. Inferences containing these correctly classified hypotheses were called easy examples. NLI models can cheat the task on these examples, as they are able to classify most of the hypotheses correctly without looking at the premise. They conclude that this hypothesis-only bias in the SNLI dataset is caused by phenomena they refer to as annotated artifacts.
Annotation artifacts are a manifestation of the heuristics adopted by crowd workers when creating the hypotheses, in order to generate hypotheses quickly and efficiently. As a result of Annotation artifacts, which are patterns in the hypotheses, hypothesis-only classifiers are able to correctly classify 67% of the data without observing the premise. These annotation artifacts give the SNLI dataset the hypothesis-only bias that is

[1] https://fasttext.cc/docs/en/supervised-tutorial.html

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Figure 1: Examples of entailment, neutral and contradiction inferences. Source: nlp.stanford.edu/projects/snli/

referred to.
Gururangan et al. (2018) presented that some of these annotation artifacts are the lexical choice and the sentence length. Lexical choice means that annotators consistently use similar or the same words in hypotheses of the same class. They found that hypotheses that are labeled as a contradiction often contain the words 'never', 'no', 'nobody', or some contradicting word such as 'sleeping' when the premise is about an activity. For the neutral class, they found that the hypotheses often contain modifiers, superlatives or purpose clauses. For the entailment class, they found that the hypotheses contain a lot of generic words and approximations, nothing too exact. The entailment class hypotheses were also often shorter and had a lot of overlap with their premise.

Contributions The thesis is organized around the research question: can we mitigate the hypothesis-only bias in the SNLI dataset? To answer this question, we will be expanding the current SNLI dataset in such a way that the annotation artifacts are more balanced over the 3 labels. Expanding the dataset in this way should reduce the effect of the annotation artifacts and therefore mitigate the hypothesis-only bias in the new augmented SNLI dataset. The augmented SNLI dataset will then be more suitable for evaluating the performance of NLI classifiers on the task of determining the label of an hypothesis based on a given premise.
Thus, the contribution of this research is a method that can be used to mitigate hypothesis-only bias in any dataset obtained in a similar manner as SNLI. Additionally, two augmented versions of the SNLI dataset, a 120k examples per case augmented dataset and a 50k examples per case augmented dataset will be created and made available[2]. These augmented datasets are better for evaluating the task performance of NLI models, because the NLI model can not use the hypothesis-only bias as a short cut.

[2]https://github.com/afra-baas/Augmenting_SNLI.git

2 Report Overview

In Section 3 Related Work, similar works to this one will be discussed and elaborated, distinctions between this work and previous works will be made.
In Section 4 Approach, the main idea of the approach will be explained, which is to balance the artifacts over the labels by generating more examples. Additionally, the attempted data augmentation techniques for achieving new automatically generated examples will be discussed.
In section 5 Results, the results will be analysed by comparing the performance of fastText on the original SNLI dataset and our augmented SNLI dataset, as well as the performance of DistilBERT on the augmented SNLI dataset.
The new relationship between hypotheses and labels, and the new sentence length distributions will also be discussed.
In Section 6 Discussion, this thesis as a whole will be discussed, as well as the strengths and weaknesses of our approach.
Additionally, direction for future work will be covered.
In Section 7 Conclusion, the research question will be answered, given the results.

3 Related Work

Debiasing Datasets One way to deal with biased datasets is to use model-level debiasing techniques. Such approaches do not alter the dataset but change the way a model handles the data. Instead of changing a model to handle biased data, datasets themselves can be debiased prior to training. Similar to this thesis, these approaches modify the data instead of altering the model/training. Some of these transformations will be discussed in this section along with their similarities and discrepancies between our approach. If a dataset is debiased, the need for model-level debiasing is decreased

as there is less bias in the data. Of course, model-level debiasing will always be needed as different models are sensitive to different biases. However, because this thesis is concerned with data-level debiasing, model-level debiasing techniques will not be discussed.

Adversarial Training Adversarial training, one of data-level debiasing techniques that will be discussed, involves training on adversarial examples that could mislead classifiers to make an incorrect prediction. These adversarial examples include character/word/sentence level perturbations of data while keeping the same label. This makes the model robust to perturbations. Belinkov et al. (2019) employed two adversarial learning techniques, adding an external adversarial hypothesis-only classifier and perturbing training examples. This showed that data augmentation can be effective for decreasing the hypothesis-only bias.

Down-sampling Down-sampling just removes the easy examples, which are inferences containing a hypothesis that can be classified correctly by a hypothesis-only classifier. However, this approach can also eliminate useful phenomena of inference that the model should learn (Liu et al., 2020).

Repeating Training Data Since the portion of unbiased samples in a dataset with a bias is smaller, another approach is to repeatedly add selected unbiased samples to balance the training data (Zhou and Bansal, 2020).

Multi-round annotation Multi round annotation is similar to adversarial training, but in this approach a human annotator creates the hypothesis intended to fool the model instead of perturbation functions (Nie et al., 2019).

Data Augmentation Data augmentation has been proven to be an effective way to tackle biases by Belinkov et al. (2019) as well as in prior research by Sharma et al. (2021) on gender bias in the SNLI dataset. They found that debiasing techniques such as augmenting the training dataset to ensure a gender-balanced dataset can help reduce the bias in certain cases. Consequently, data augmentation will also be used in this thesis to mitigate hypothesis-only bias in the SNLI dataset. Wei and Zou (2019) presented that easy data augmentation (EDA) techniques could already give a good boost to the performance of text classification. EDA is a set of 4 simple universal augmentation techniques: synonym replacement, random insertion, random swap, and random deletion.

4 Approach

4.1 Methodology

This thesis will attempt a data-level debiasing approach by applying data augmentation and distinguishes itself by generating new inferences that intentionally have a different label. This way the the structure of the dataset is changed. Each premise in the SNLI dataset has three hypotheses with a different label. Therefore each premise can relate to three labels, but an hypothesis only relates to one. This gives NLI models the opportunity to link the annotation artifacts, present in the hypotheses strongly to a particular label.
It is considered and taken into account that the cause of this bias is the fact that an annotation artifact is linked to only one label (Gururangan et al., 2018; He et al., 2019). Consequently, this is aimed to be changed with data augmentation. The generated data will change the structure by making most of the hypotheses relate to more than one label, ideally three. By generating new examples all the heuristic distributions are also balanced at once. Lexical choice nor sentence length will be linked to only one label. Important to note is that we will only be changing and augmenting the train dataset.
Consider a premise and hypothesis pair labeled entailment. From this inference the hypothesis is taken and a new premise is generated, creating a new premise and hypothesis pair (inference) that can be labeled contradiction. Then, another premise is generated that will lead to a premise and hypothesis pair that can be labeled neutral.
To make a hypothesis relate to 3 labels: neutral, contradiction and entailment depending on the premise, 6 types of label changing cases have to be generated. Figure 2 gives a visualisation of the 6 cases and the structure of the original SNLI and the augmented SNLI. This way every pattern can be linked to more than one label. Making it more likely that the NLI classifier will need to look at the premise and use actual reasoning to label the hypothesis, and therefore label the inference. It could still find another short-cut, but it will no longer able to use hypothesis-only bias as a short-cut.
FastText will be used as a hypotheses-only classifier to check if the portion of correctly classified hypotheses has decreased. The same method will be applied to check if no premise-only bias has been created in the augmented dataset. Additionally, the NLI classifier, DistilBERT (Sanh et al., 2019) is fine-tuned on the augmented SNLI dataset. In order for this approach to be considered a success, the hypothesis-only classification accuracy with fastText should decrease significantly while the performance of DistilBERT should not decrease significantly compared to the performance it has when fine-tuned on the original SNLI.

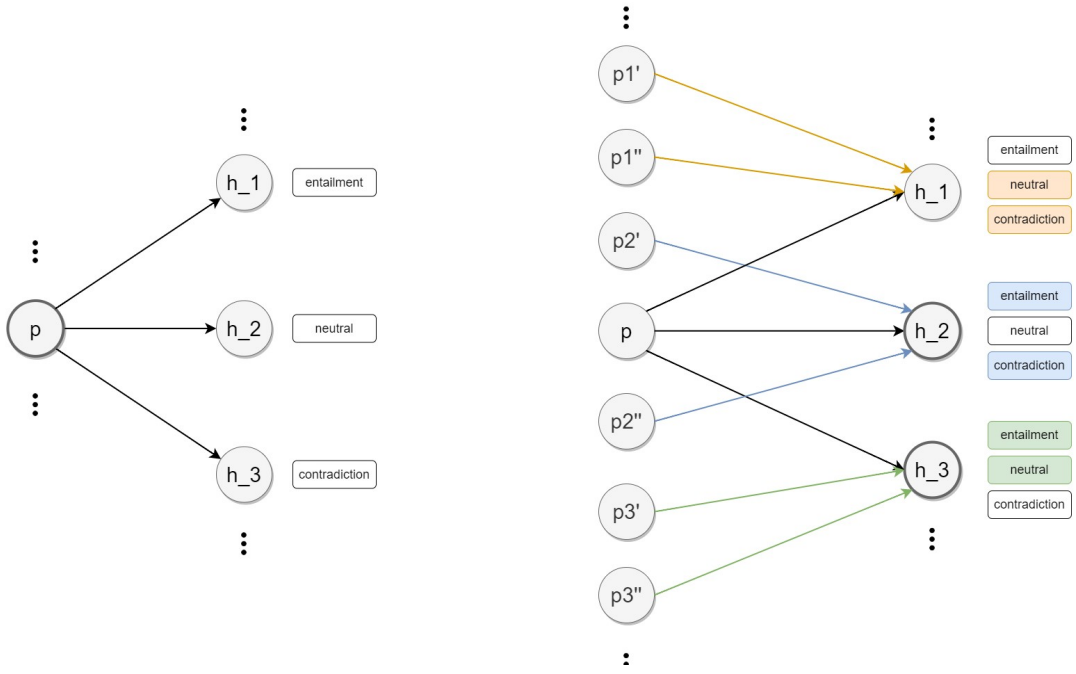

Figure 2: The structure of SNLI dataset (on the left) and the structure of the augmented SNLI dataset (on the right); The colors indicate the new cases.

4.2 Data augmentation

In the section 3 Related Work, some data augmentation techniques were discussed that served as inspiration for the perturbation functions used to obtain each of the 6 cases, see figure 3. Perturbation functions take a hypothesis[3] as input and the output is a perturbed version of this hypotheses, that can be used as the new premise to obtain an inference with a specific label. The frameworks that can be used for data augmentation and our implementation of them will be discussed in this section.

Data Augmentation Frameworks From CheckList Frameworks (Ribeiro et al., 2020) we used the CheckList RoBERTa suggestions template function for word insertion. CheckList RoBERTa can be used to inserts a word into a sentence, by putting a '{mask}' where the new word is wanted. This function was used such that before every noun and verb, there is a 50% chance that a word will be inserted. CheckList negate, which is another predefined function that negates sentences, was also applied for sentence negation.
We used WordNet[4] (Miller, 1995) for swapping a noun, a verb or an adjective with either its antonym, hyponym or hypernym[5]. WordNet is a large lexical database of English words, in which word relations such as antonyms, hyponyms, hypernyms or meronyms can be found.

Implementation The experiment was executes with the help of Google Colab[6], which allows anyone to write and execute python code in their browser, as well as share the (Jupyter) notebook[7] with others similar to Google Docs or Sheets. It also offers limited access to GPUs an TPUs for free.
Before starting with swapping nouns, verbs and adjectives it should be noted that WordNet uses synsets not only to store synonyms of a word, but also to group words with the same meaning together. Each word in the synset shares the same sense of the word. When working with WordNet it is important to know the sense of the word. For this reason the sense of the most frequent nouns, verbs and adjectives were checked and saved. Additionally, to avoid the introduction of unsuit-

[3]The only hypotheses considered in this approach are those that had a gold-label: 'entailment', 'contradiction', 'neutral'. The inference labeled undetermined ,'-' were not considered.

[4]https://wordnet.princeton.edu/

[5]Adjectives were only swapped with an antonym and in a few cases, since there were only a few adjectives with antonyms

[6]https://colab.research.google.com/notebooks/intro.ipynb?utm$_source = scs - index$

[7]The code is executed in Colab notebooks, which are interactive programming environments, similar to Jupyter Notebook.

old premise:	"A person on a horse jumps over a broken down airplane."
(1) new premise:	"A person is at a diner, ordering an omelette or A woman with a red helmet on is jumping."
(2) new premise:	"A person is at a diner, ordering an omelette and A woman with a red helmet on is jumping."
hypothesis:	"A person is at a diner, ordering an omelette."

Table 1: A new neutral example (1), The case of contradiction to neutral; A new neutral example (2); The case of contradiction to entailment

able words[8] when using the swap function, a separate set was made containing all the nouns, verbs and adjectives in the training data and a swap was only allowed with a word that already occurred in the training data.
Furthermore, functions for word deletion, changing quantifiers e.g 'some' becomes 'all', changing the tense of a verb e.g from present to past were also considered. These led to the generation of very diverse [9] new premises, but it often also occurred that the sentences were no longer correct or the label did not change in the predicted way. For that reason these latter functions were later removed.
We work with cases that change a hypothesis of one label to another, when related to the new premise. That is why it is very important that the label of the hypothesis changes in the way predicted by the case it is in, to allow for automatic labeling later. The grammar of the sentences was also monitored, by doing a manual check of about 20 examples of each case, after every 5k new examples. Besides the typo's that were already in the data, or the occasional wrong sense of the word, the sentences are grammatically correct and decently natural
Generating a premise that reliably led to a neutral inference was difficult to obtain with the mentioned functions, which are word swaps, word insertions, sentence negation. For this reason, a rule was made that uses a logical approach to generating neutral inference examples. The 'H or S' schema was used[10]. After each hypothesis the string 'or' and any other premise was added. With this rule the hypothesis is always neutral, since either the part before (the hypothesis) or after the 'or' (a random premise) can be true. The first new premise in table 1 is an example.
A logical rule was also used for generating entailment examples, because the running time of insertion and swapping was long. With this additional manner of generating entailment examples, one could cut down on this running time. The the 'H and S' schema was used, after each hypothesis the string 'and' and any other premise[11] was added. The second new premise in table 1 is an example.

4.3 The 6 Cases Explained

This section explains what the perturbation function does in all 6 cases, a visualization is given in figure 3 and it shortly displays the only the most important part of the functions does.
For each case 20 random sentences were taking to check how accurately the labels have been changed in the intended way. The results are in table 2. The accuracy was decent and therefore the generated examples could all be labeled with the latter label of the case it is in.

Case	accuracy
Entailment to contradiction	17/20
Entailment to neutral	20/20
Neutral to contradiction	17/20
Neutral to entailment	19/20
Contradiction to neutral	20/20
Contradiction to entailment	16/20

Table 2: The accuracy of the different cases.

Case 1: Entailment↦Neutral For this case all the inferences with a gold-label entailment are filter out and put in a list. From each of these entailment inferences we take the hypothesis and give it to the perturbation function. With this function we want to generate a new premise that causes the hypothesis to no longer be an entailment, but neutral. The function will take the hypothesis and attach to it the string 'or' and any premise in the train dataset, except the premise belonging to the given hypothesis. An example is the first new premise given in table 1.

[8] an example of an unstable word is the fact that one of the hyponyms for 'man' is 'bull' and one for 'woman' is 'nymph'

[9] With diverse we mean the the sentences could be perturbed in more ways.

[10] in the schemas, S refers to any other premise in the train dataset that does not belong to the hypothesis.

[11] DistilBERT is uncased 'distilbert-base-uncased', it does not depend

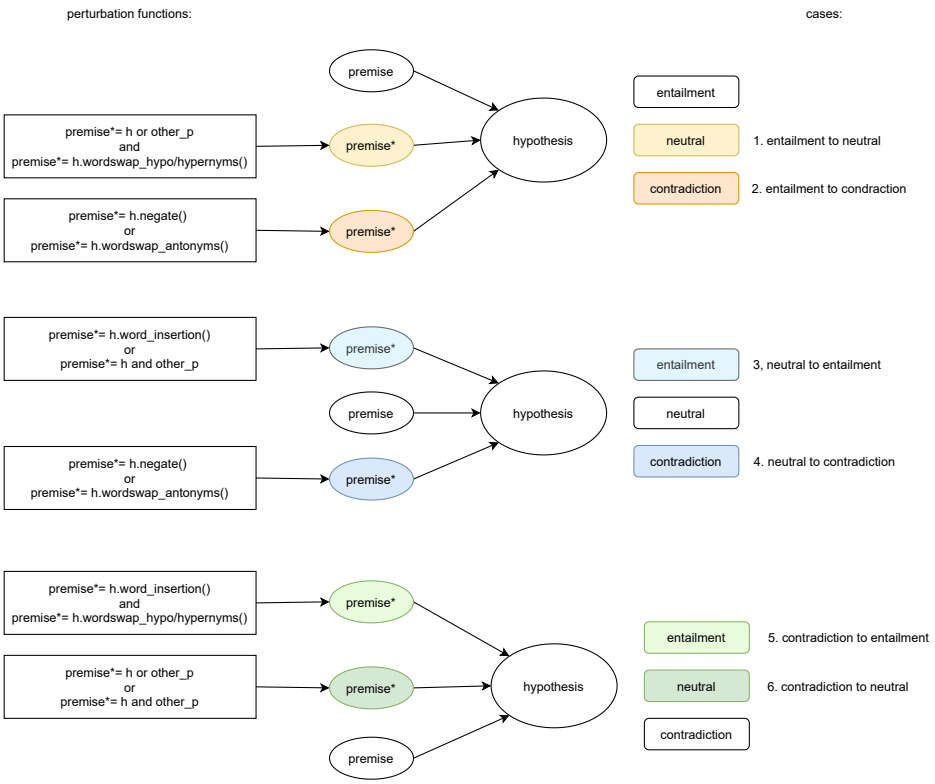

Figure 3: the perturbation function of the 6 cases

Case 2: Entailment↦Contradiction For this case all the inferences with a gold-label entailment are taken and from each of these entailment inferences we take the hypothesis and give it to the perturbation function. With this function we want to generate a new premise that causes the hypothesis to no longer be an entailment, but a contradiction. This function will take the hypothesis and swaps the nouns and verbs with its antonyms. If a verb was not swapped with one of its antonyms, then the CheckList negate will be applied to it. CheckList RoBERTa is also used to insert a word, with a 50% chance before every noun and verb. After a word insertion we attempt to negate the sentence again, if CheckList negate could not negate it before. An example is given in table 3.

old premise:	”Two women who just had lunch hugging and saying goodbye.”
new premise:	”There certainly are not two female in this picture.”
hypothesis:	”There are two woman in this picture.”

Table 3: example of case 2

Case 3: Neutral↦Entailment For this case, all the hypotheses with the gold-label neutral are filtered out and the hypothesis is given to the perturbation function to generate a new premise that will change the hypothesis label to entailment. There is a 30% [12] chance the function will use the ’H and S’ scheme to generate an example and use CheckList RoBERTa to insert a word, with a 50% chance before every noun and verb. Otherwise, the function takes the hypothesis and swaps the nouns and verbs with its hyponym or hypernym and uses CheckList RoBERTa to insert a word, with a 50% chance before every noun and verb. An example of the type of sentence generated in 70% of the cases is given in table 4. When the ’H and S’ scheme is used the sentences look very similar to the second new premise in table 1.

Case 4: Neutral↦Contradiction For this case all the hypotheses with the gold-label neutral are collected. From each of these inference the hypothesis is taken and given to the perturbation function that is used to generate a new premise that will change the hypothesis la-

[12] By making a trade-off between speed of the function, and a preference for generating the sentence with word swaps and insertions, we then decided to generate 30% of the entailment sentences with the ’H and S’ scheme.

old premise:	"A man in a baseball hat and sunglasses watching an event in a crowd."
new premise:	"A boyfriend is watching a charity baseball event."
hypothesis:	"A man is watching a baseball event."

Table 4: example of case 3

old premise:	"a man wearing a robe stands next to a table full of various cheeses"
new premise:	"A adult devouring coconut bread butter at daily."
hypothesis:	"A man eating bread butter at daily."

Table 6: example of case 4

bel to contradiction. The function takes the hypothesis and swaps the nouns and verbs with its antonyms. If a verb was not swapped with one of its antonyms, then the CheckList negate will be applied to it. CheckList RoBERTa is also used to insert a word, with a 50% chance before every noun and verb. After a word insertion we attempt to negate the sentence again, if CheckList negate could not negate it before. It this case we also left the function that changes a quantifier, e.g 'all' to 'some', if present in the sentence, if a noun was not swapped with one of its antonyms. An example is given in table 5.

old premise:	"a boy jumping off a swing"
new premise:	"The male is not having absolute fun."
hypothesis:	"The boy is having fun."

Table 5: example of case 4

Case 5: Contradiction↦Entailment For this case we filtered out all the hypotheses with the gold-label contradiction and each hypothesis was given to the perturbation function to generate a new premise that will change the label of the hypothesis to entailment. There is a 30% chance the function will use the 'H and S' scheme to generate an example and use CheckList RoBERTa to insert a word, with a 50% chance before every noun and verb. Otherwise, the function takes the hypothesis and swaps the nouns and verbs with its hyponym or hypernym and uses CheckList RoBERTa to insert a word, with a 50% chance before every noun and verb. An example of the type of sentence generated in 70% of the cases is given in table 6. When the 'H and S' scheme is used the sentences look very similar to the second new premise in table 1.

Case 6: Contradiction↦Neutral For this case all inference with a gold-label contradiction are taken and the hypothesis is given to the perturbation function. To generate a new premise that will change the label to neutral, the function takes the hypothesis and attaches the string 'or' to it, as well as a random premise from the training data that is not the premise of this hypothesis. An example is the first new premise given in table 1.

4.4 New Data distribution

Distribution of hypotheses relating to 1,2 or 3 labels It should be noted that sometimes the sentence came out of the function unchanged. If the sentence is unchanged, then no new inference example is generated of that case for that hypothesis. As a result, not every single hypothesis will be relating to 3 labels.

The augmented data did not generate evenly for all cases, because the perturbation function for making a neutral inference example could always output a changed sentence, generate an example of neutral inference. However, entailment and contradiction could not always generate an inference example, given a certain hypothesis as input. Besides that the perturbation function for making a contradiction inference example was the slowest, because the CheckList negate function sometimes takes a while. Thus, we decided to stop the generation of new contradiction examples after about 120k examples. To keep the distribution even, there is one augmented dataset consisting of the SNLI train data and 120k examples of each case, leading to an augmented dataset of 1270k examples called the 120k augmented dataset. There also another smaller augmented dataset[13] consisting of the SNLI train data and 50k examples of each case, leading to an augmented dataset of 850k examples called the 50k augmented dataset. We decided to work with the 50k augmented SNLI dataset, due to limited computational resources when using google colab and time constraints. Table 7 shows the amount of hypotheses with the label A, which is either entailment, neutral or contradiction, present in the training data. Each label is 33% of the training data, which has a total size of 550152 examples. Figure 6 shows the distribution of how many hypothesis relate to 1,2, or 3 labels, in 50k augmented dataset and in the original SNLI dataset.

[13] both the 120k per case augmented dataset and the 50k per case augmented dataset will be made available at https://github.com/afra-baas/Augmenting_SNLI.git.

Case (A to B)	Sentences with label A	Sentences with label B*
Entailment to contradiction	183416	132367 (72%)
Entailment to neutral	183416	183416 (100%)
Neutral to contradiction	182764	142038 (78%)
Neutral to entailment	182764	169212 (93%)
Contradiction to neutral	183187	183187 (100%)
Contradiction to entailment	183187	181775 (99%)

* percentage of hypotheses in that had label A, but now have label B

Table 7: The number of generated sentences for each case.

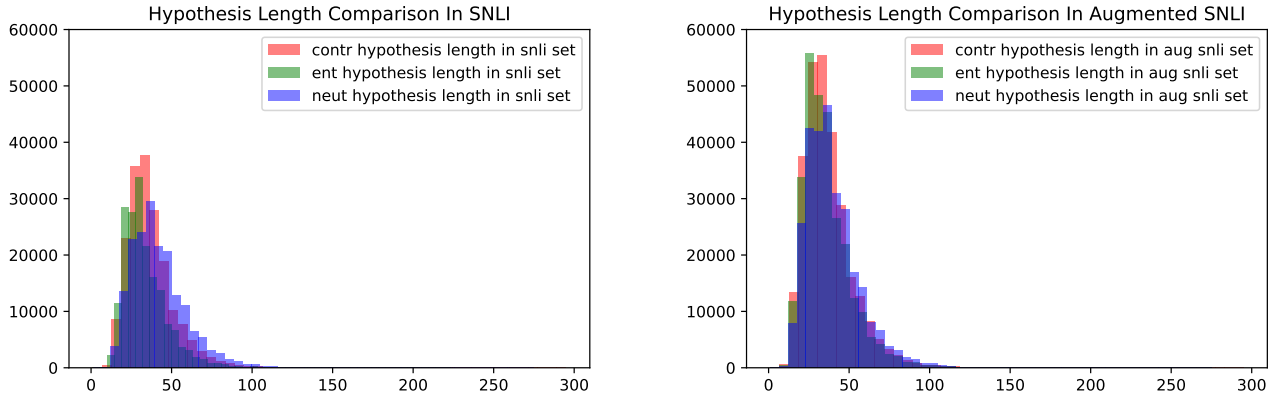

Figure 4: Hypothesis length comparison in SNLI and augmented SNLI

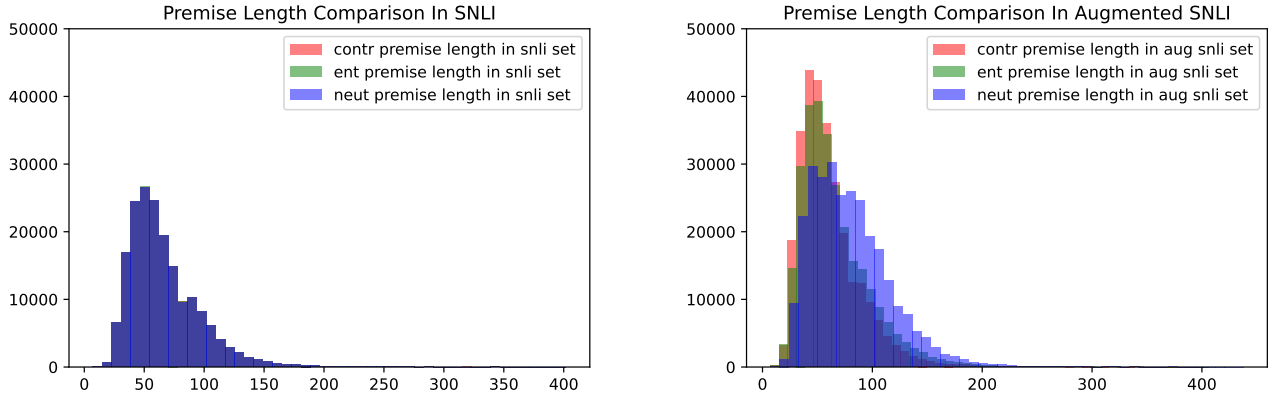

Figure 5: Premise length comparison in SNLI and augmented SNLI

Sentence length distribution The heuristic sentence length, discussed in section has also been balanced over the different labels. Figure 4 displays that the distribution of hypothesis length used to differ between the labels, but in the augmented SNLI dataset the hypothesis length is also balanced over the labels, the distribution of hypothesis length is similar for the labels. Figure 5 shows that the distribution of premise length was similar for contradiction and entailment, but neutral premises were often short. In the augmented SNLI dataset the premise length is also balanced over the labels, the distribution of premise length is similar for the labels.

5 Results

The fastText model, as a hypothesis-only classifier achieved an accuracy of 0.665 on the test set, when trained on the hypotheses of the original SNLI dataset. This is achieved with a learning rate of .2, Ngram of 2

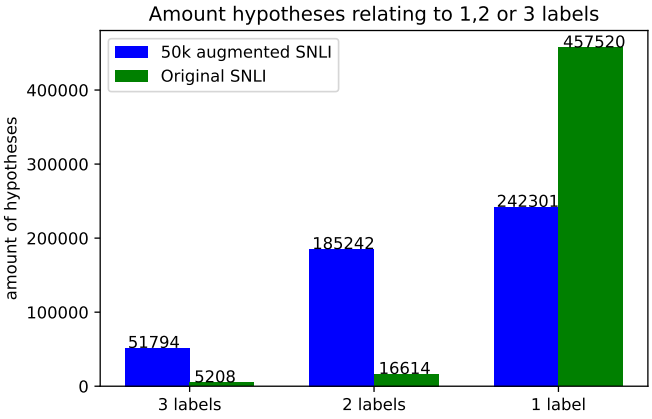

Figure 6: Amount of hypotheses relating to 1,2 or 3 labels in augmented SNLI and the original SNLI.

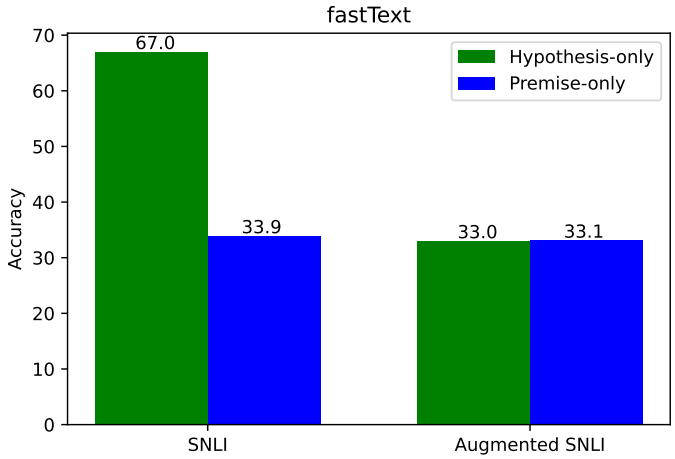

Figure 7: Hypothesis-only classification results and premise-only classification results with fastText.

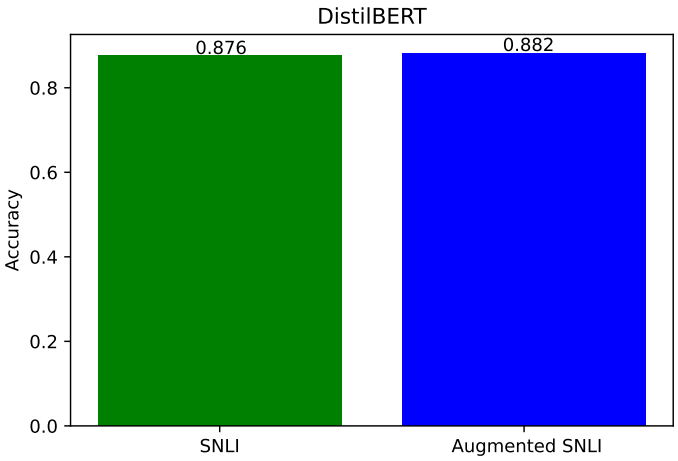

Figure 8: The accuracy of DistilBERT fine-tuned on SNLI dataset and fine-tuned of augmented SNLI dataset.

and 5 epochs.

The fastText model, as a hypothesis-only classifier, achieved an accuracy of 0.33 on the test set when trained on the hypotheses of the augmented SNLI dataset. The results of the fastText model for the premise-only classification are displayed in figure 7 and show that there did not arise any premise-only bias. The accuracy of the fastText model, as a premise-only classifier, trained on the premises of the original SNLI dataset was 0.339 and the accuracy when trained on the premises of the augmented dataset was 0.331. FastText is a good text classifier for testing hypothesis-only bias, but is not good for evaluating inferences. The accuracy of the fastText classifier on the original SNLI dataset, if given the premise and hypothesis as one string is only 0.60. This 60% is pretty low, when compared to state-of-the-art NLI models, such as BERT that can achieve an accuracy up too 90% (Talman and Chatzikyriakidis, 2019). The accuracy scores of the DistilBERT[14] classifier are also a lot higher, 0.87 for SNLI and 0.88 for augmented SNLI, as shown in figure 8. DistilBERT was fine-tuned on the development dataset and with a batch-size of 32, steps of 5000, learning rate of 2e-5, weight decay of 0.01, and 1 epoch.

6 Discussion

We will now reflect on our work and the first improvement is about hypotheses appearing more than once. When determining how many labels a hypothesis relates to, we discovered that some hypotheses appear more than once in the original SNLI dataset. This seems to be because crowd workers sometimes reuse a created hypothesis. They reuse a hypothesis for different premises, sometimes for the same label and sometimes for another label. Thus, some hypotheses already relate to more than one label. More importantly it should be noted not to filter on just gold-label, but to also remove double hypotheses. For the cases where a hypothesis was reused with another label, there could be a check before the perturbation function, to make sure the hypothesis is not already in the list of hypotheses for which the function is going to make a new example. Without this check it would just create another inference example, while the hypothesis already relates to that label. An example of a reused hypothesis in the original SNLI dataset is given in figure 9.

[14]DistilBERT was find-tuned using steps instead of epochs, because epochs take too long to run in google colab.

	gold_label	sentence1	sentence2
128008	entailment	A DJ wearing headphones and mixing some music.	A DJ is playing music.
250281	neutral	A man in a gray shirt is operating the soundboard for a social gathering.	A DJ is playing music.
271698	entailment	A DJ is spinning tracks at a club.	A DJ is playing music.

Figure 9: A hypothesis occurring multiple time in the SNLI dataset

Weaknesses Ultimately, Not all hypotheses could be given a relation to 3 labels, see figure 6. The perturbation functions were sometimes unable to generate a new premise and the sentences have some issue. The labelling accuracy could also be improved if the perturbation functions are refined. The perturbation functions could use improvement, we elaboration on this is in the paragraph 'Future work'. Additionally, the performance of DistilBERT is only evaluated the 50k augmented dataset due to lack of resources.

Strengths However, the results show that even if not all hypotheses relate to more than one label, the hypothesis-only bias can still be mitigated. The performance of the NLI model DistilBERT is also good even when the sentences have some issues, which will be discussed in the paragraph Error analysis 6. Perhaps if the proposed method is executed again and with more resources, there might even be a clear indication that the NLI model improves when trained on the augmented dataset. Figure 8 shows that the accuracy of DistilBERT increased a little and it would be logical that if a bias is removed, the model might be able to bypass a hinder and learn the task of NLI better.

Error analysis There are some mistakes that slipped through that could be fixed by writing the functions better. These are displayed in tables 8, 9, 10, 11 and 12.
When using CheckList negate, we also noticed there are cases where a sentence does not get negated. For instance, the sentence "A man is using a hammer" does not get negated. When 'using' is replaced with 'walking', "A man is walking a hammer" then the sentence does get negated and it becomes "A man is not walking a hammer".
CheckList negate also uses 'not' by default and that sometimes causes unsuitable sentences like "there were not people at the party". For some of these cases were 'not' appeared before a noun, we hard coded that 'no' comes before the noun instead. We only did this for a list of words were the problem was observed a lot, namely 'people', children' and 'schools'. However, this list should be expanded or the reason for the error can be identified and addressed. There are still a lot of sentences that get negated by putting 'not' before the noun instead of 'no'. The reason for these two issues when negating the sentence with CheckList negate are beyond this thesis.

old premise:	"Two children are laughing in the grass."
new premise:	"The peanuts are taking a bath."
hypothesis:	"The children are taking a bath."

Table 8: A mistake in the contradiction to entailment case

old premise:	"Guys riding skateboards in a skate park."
new premise:	"A poodle and a cat cat are playing. "
hypothesis:	"A dog and a cat are playing."

Table 9: A mistake in the contradiction to entailment case

old premise:	"A group of people sitting in an ornate church are focused on a clergyman speaking into a microphone."
new premise:	"... Everybody here is atheist. "
hypothesis:	"Everybody here is atheist."

Table 10: A mistake in the contradiction to entailment case

Future work For future work there is a logical, rule-based text classifier LangPro[15] (Abzianidze, 2017) that can be used to check the cases where the new inference example has to become an instance that can be labeled as contradiction or entailment. If LangPro classifies an inference as entailment or contradiction, then it is certain

[15] https://naturallogic.pro/LangPro/

old premise:	”Men who are walking in front of park benches.”
new premise:	”The forces haven’t come to the park to play frisbee.”
hypothesis:	”The men have come to the park to play frisbee. ”

Table 11: A mistake in the neutral to contradiction case

old premise:	”A couple wearing pink and purple hoodies have their arms across each other’s backs as they walk down the sidewalk.”
new premise:	”Two sisters Please are not going on a walk together.”
hypothesis:	”Two sisters are going on a walk together.”

Table 12: A mistake in the neutral to contradiction case

that this inference is an entailment or contradiction. This would be a nice way of checking what percentage of the new examples are indeed an entailment or contradiction and would require less manual checks. When LangPro classifies an inference as entailment and contradiction is it certain, but if LangPro classifies an inference as neutral it could still be entailment or contradiction. LangPro also cannot solve many diverse problems as it is rule-based. Thus, manual checks (or another classifier) will still be needed for the inferences LangPro classifies as neutral.

Future work could also use a different labeling method, that does not require the perturbations function to change the label in a predicted way and that will allow them to generate more diverse premises. For example, crowd sourcing can be used to for labeling the new inference examples.

SNLI only contains a fraction of what NLI is, it has a particular definition of inference. So generating diverse premises and then filtering out sensible ones and labeling them with the help of crowd sourcing can introduce different phenomenons and a broader definition of inference. This would also ensure that less unsuitable sentences with mistakes stay in the augmented dataset. When people are asked to make inferences, they adopt heuristics for efficiency unlike end users will. Thus, it would be better to generate new inference with e.g perturbation functions and let people filter and label the inferences.

There are still quite a lot of issues with the newly generated sentences. An interesting direction that can also be taken, is to re-evaluated the sentences of the augmented SNLI dataset and question whether the quality of the sentences is good enough for it to be justified to train a classifier on these sentences. What defines the quality of a sentence and what makes it good for training a classifier can then also be determined. Additionally, there could be an attempt to make better functions for automatically generating sentences and investigate how the quality of the generated sentences in a augmented dataset affects an NLI model. Then a supported argument could be given of whether it is justified to train an NLI model on a dataset with sentence of less good quality.

7 Conclusion

We used perturbation functions to generate new premises that can change the label of the hypotheses. This was to balance the annotation artifacts over the different labels, since this would lead to less hypotheses that relate to only one label. The new distribution of the amount of hypotheses relating to 1,2 or 3 labels is displayed in figure 6 and shows that half of the hypotheses now relate to only one label.

The results in figure 7 show that fastText classifies less hypotheses correctly, without looking at premise when trained on the augmented dataset compare to when trained on the original dataset. Therefore, it can be concluded that the hypothesis-only bias in the SNLI dataset can indeed be mitigated by balancing the artifacts in the hypotheses over all 3 labels. The cause for the hypothesis-only bias in the SNLI dataset was the relation between annotation artifacts and the label. Important to note is that it was not even necessary for every single hypothesis to relate to 3 labels. Figure 6 shows that 51%[16] of the hypotheses in the augmented SNLI dataset still relate to one label, but the hypothesis-only bias has been mitigated.

The results in figure 8 show that a NLI model trained on the augmented dataset still achieves a competitive accuracy, thus it can be concluded that the augmented dataset is good and still captures some definition of inference. This approach and the functions used are simple, effective and can be used on any inference dataset with annotation artifacts relating to only one label. The structure of the dataset is very important in controlling what NLI models actually learn. Machine learning finds the most efficient way to make predictions by looking at the patterns in the dataset. If the dataset is not structured carefully it will learn an unwanted pattern. When it comes to the data-level debiasing, using data augmentation with the intend of changing the structure of the dataset to remove a relation that caused a bias, has been proven effective.

[16] hypotheses relating to 1 label / hypotheses relating to 1,2 or 3 labels = 242301 / 479337=0.51

8 Acknowledgments

Throughout the writing of my thesis I have received a great deal of support and assistance. I would like to thank my supervisor Lasha Abzianidze, whose expertise was invaluable in formulating the research question and methodology. I would also like to thank my friends and family for their support.

References

Abzianidze, L. (2017). LangPro: Natural language theorem prover. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 115–120, Copenhagen, Denmark. Association for Computational Linguistics.

Belinkov, Y., Poliak, A., Shieber, S. M., Durme, B. V., and Rush, A. M. (2019). On adversarial removal of hypothesis-only bias in natural language inference. *CoRR*, abs/1907.04389.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

He, H., Zha, S., and Wang, H. (2019). Unlearn dataset bias in natural language inference by fitting the residual. *CoRR*, abs/1908.10763.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Liu, T., Zheng, X., Chang, B., and Sui, Z. (2020). Hyponli: Exploring the artificial patterns of hypothesis-only bias in natural language inference. *CoRR*, abs/2003.02756.

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

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2019). Adversarial NLI: A new benchmark for natural language understanding. *CoRR*, abs/1910.14599.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Durme, B. V. (2018). Hypothesis only baselines in natural language inference. *CoRR*, abs/1805.01042.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (2020). Beyond accuracy: Behavioral testing of NLP models with checklist. *CoRR*, abs/2005.04118.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Sharma, S., Dey, M., and Sinha, K. (2021). Evaluating gender bias in natural language inference. *CoRR*, abs/2105.05541.

Talman, A. and Chatzikyriakidis, S. (2019). Testing the generalization power of neural network models across NLI benchmarks. In Linzen, T., Chrupala, G., Belinkov, Y., and Hupkes, D., editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 85–94. Association for Computational Linguistics.

Wei, J. W. and Zou, K. (2019). EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguistics*, 2:67–78.

Zhou, X. and Bansal, M. (2020). Towards robustifying NLI models against lexical dataset biases. *CoRR*, abs/2005.04732.