Exploring data biases in document-level natural language inference dataset

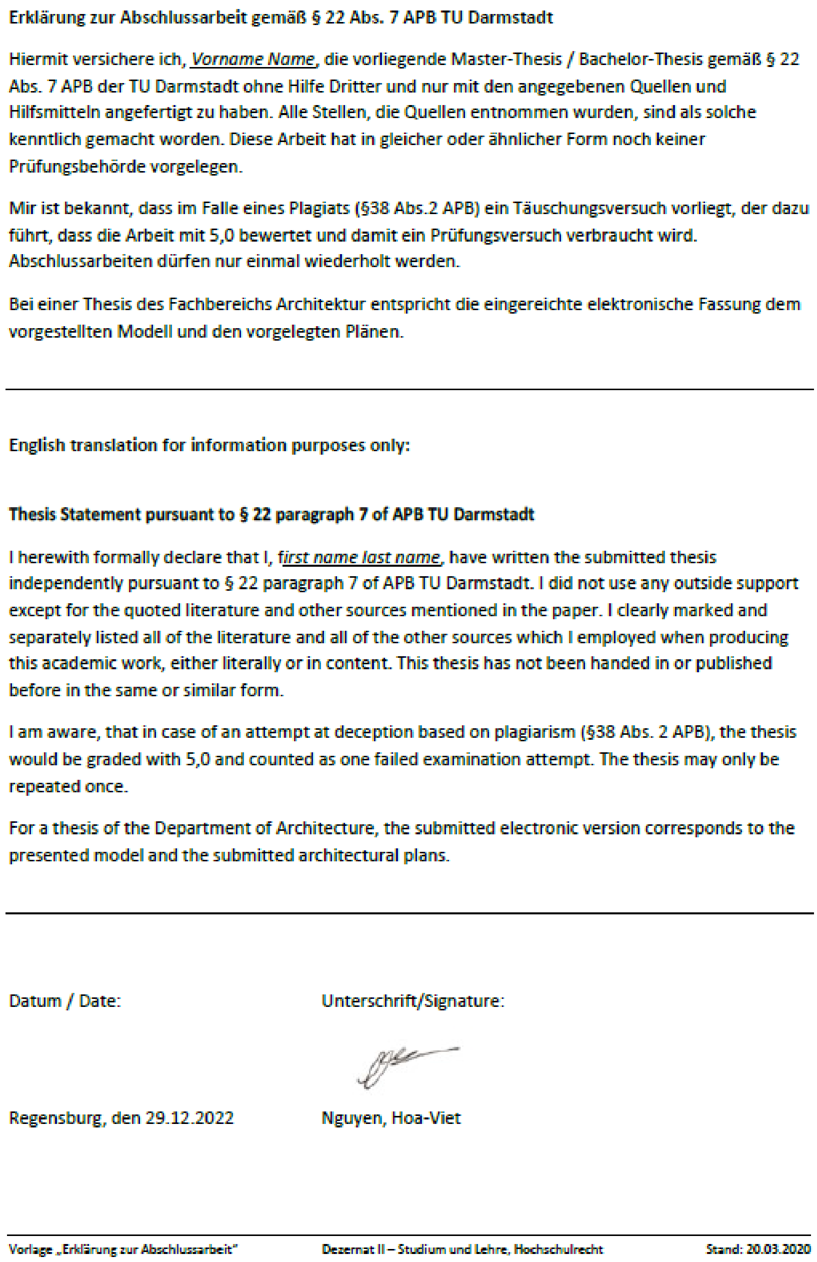
A study on bias discovery and bias mitgation techniques on DocNLI dataset
Master thesis in Computer Science by Hoa-Viet Nguyen
Date of submission: December 29, 2022

1. Review: Prof. Dr. Iryna Gurevych
2. Review: Irina Bigoulaeva M.Sc.
3. Review: Thy Thy Tran PhD
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

UBIQUITOUS
KNOWLEDGE
PROCESSING

Computer Science
Department
Ubiquitous Knowledge
Processing Lab



Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, *Vorname Name*, die vorliegende Master-Thesis / Bachelor-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

English translation for information purposes only:

Thesis Statement pursuant to § 22 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, *first name last name*, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.

I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.

For a thesis of the Department of Architecture, the submitted electronic version corresponds to the presented model and the submitted architectural plans.

Datum / Date:	Unterschrift/Signature:
Regensburg, den 29.12.2022	Nguyen, Hoa-Viet

Vorlage „Erklärung zur Abschlussarbeit" Dezernat II – Studium und Lehre, Hochschulrecht Stand: 20.03.2020

Abstract

Document-level NLI is a critical capability for many downstream tasks. However, works about document-level biases in the NLI dataset appear less frequently in the NLP literature, in contrast to the well-studied sentence-level biases in the NLI dataset. In this study, we attempted to find potential biases in the document-level dataset DocNLI and examine the effected of these biases on the performance of NLI models trained on it. An adversarial dataset is created to reveal the weaknesses of the models trained on the biased dataset. In addition, we compare state-of-the-art bias reduction techniques previously used to reduce sentence-level bias from both data-centric and model-centric perspectives. Our results suggest that models trained on DocNLI suffer from syntactic heuristic learning due to dataset bias. Additionally, sentence-level bias reduction techniques can be transferred to the same document-level task.

Acknowledgement

I would like to sincerely thank Irina Biguolaeva and Thy Tran for their dedicated guidance during the writing time of this thesis, their intermediate feedback for constant improvements and last but not least for their patience with me. I am profoundly grateful to be given this opportunity to work under your supervision. They really contributed to my great experience as a student of the Masters of Informatics program at the Technical University of Darmstadt. I want to express my appreciation to all my friends Stefan Löffler, Stella Maris Sibarani and Van Hong Le for their valuable feedback on my first draft of the thesis. Lastly, heartfelt thanks go to my parents. Without their love and support, this work would not have been possible.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

The development of a intelligence system that owns reasoning skills is a longstanding goal of artificial intelligence, also refers to Natural Language Understanding (NLU). Applications of NLU systems in our daily life are wide-ranging from question-answering, fake news detection, multiple choice reading comprehension, and text summarization systems. However, these systems are far from achieving human-level reasoning skills, often lacking of the ability to perform different types of contextual reasoning.

Over the past years, a promising research direction that has gained more and more attention is building a general Natural Language Inference (NLI) system to solve many tasks in NLU through recasting subtasks in NLU into a NLI problem. For example, Mishra et al. (2021) attempt to transform the task of question-answering and checking-factual consistency of text summaries into NLI problem. Trivedi et al. (2019) also propose to cast Multihop Commonsense Reasoning into NLI. This lines of research support the direct use of a general NLI models on downstream tasks.

While most of the NLI problems being recasted from down-stream tasks are often performed at the passage-level or document-level, a large body of recent benchmark NLI datasets are at sentence-level, in which both the premise and the hypothesis consist of a single sentence. Document-level NLI and passage-level, in which the premise and hypothesis can span multiple sentences or even paragraphs, remains relatively less explored despite its relevance for many downstream tasks.

There has been a few document-level NLI datasets constructed to fill up this gap such as ContractNLI (Koreeda and Manning, 2021) and DocNLI (Yin et al., 2021). ContractNLI is a domain-specific dataset of legal contracts, covering 670 non-disclosure agreements (NDAs) of 17 types. DocNLI is a multi-domain dataset sourced from four different NLP benchmarks: adversarial NLI (ANLI) (Nie et al., 2020), the question answering benchmark SQuAD (Rajpurkar et al., 2016) and three summarization benchmarks: DUC20011, CNN/DailyMail (Nallapati et al., 2016), and Curation (Curation, 2020).

The existence of datasets alone, however, is not enough to foster improvement in document-level NLI. A potential pitfall of this task is shallow heuristics learning, where a model learns spurious correlations from the training data rather than the desired complex reasoning abilities. This is due to biases being present in the data. We define "bias" as the tendency of a dataset to overrepresent data samples that render a specific type of heuristic. For example, Gururangan et al. (2018) show that the popular sentence-level datasets SNLI and MNLI exhibit a "hypothesis-only" bias, since they contain many data samples that allow a model to predict the correct NLI label using only the hypothesis, without considering the given premise. Additionally, McCoy et al. (2019) suggest that the majority of samples within SNLI and MNLI embody shallow heuristics between the premises and hypotheses that are exploited by the NLI models such as lexical overlapping, subsequence heuristics, and constituent heuristics, each of which can potentially produce additional biases. The models trained on such biased data can perform well on the test set, but exhibit unintuitive and undesirable behavior on real-world data. The document-level DocNLI dataset is also known to be biased, resulting, for example, in a RoBERTA-based NLI model developing the -only" bias Yin et al. (2021).

While previous works have focused on making NLI models less susceptible to learning spurious heuristics, thus addressing the problem from the model side (Clark et al. (2019); Karimi Mahabadi et al. (2020); Utama et al. (2020b); Sanh et al. (2021)), it is equally important to address the problem from the dataset side. Several studies show that the prevalence of “hypothesis-only” biases in various sentence-level NLI datasets results from preventable flaws in the data annotation process, such as leaving out gender and number information when constructing entailed hypotheses and predominantly using negation when constructing contradictory hypotheses (Gururangan et al. (2018); Glockner et al. (2018); McCoy et al. (2019)). Currently, however, there is a lack of work that deals with biases in document-level NLI datasets.

Motivated to close this gap, our goal is to detect biases in existing document-level NLI datasets, provide concrete evidence for these biases and propose strategies for removing them. Our work aims to provide higher-quality versions of these datasets, in order to encourage models to acquire proper reasoning skills during training, even if the models lack self-debiasing mechanisms to avoid learning shallow heuristics.

1.2 Research questions

In this thesis, we investigate the above-mentioned aspects and formulate them into our research questions (RQs):

1. Do existing document-level NLI datasets contain biases?
2. To what extent do existing NLI systems trained on these datasets learn heuristic rules rather than the intended reasoning abilities to make a correct prediction?
3. Which kinds of debiasing methods for document-level NLI datasets are effective?

1.3 Methodology and Contributions

Our main contributions are the following:

1. We systematically investigate biases in a document-level level dataset DocNLI.
2. We introduce a novel adversarial dataset that measures the robustness of models trained on existing document-level NLI datasets. We test and expose the vulnerability of the existing NLI models using the adversarial dataset.
3. We experiment with variants of existing dataset debiasing methods from sentence-level NLI to document-level NLI. We measure and compare the effectiveness of these adapted debiasing methods on document-level NLI.

The following chapters of the thesis is organized as follows. In chapter 2, we introduce different bias forms in natural language inference datasets and various debiasing methods to over this problem in the literature. We reproduce the baseline of DocNLI in the Chapter 3. In chapter 4, we carry out experiments to quantitatively identify the potential bias in the DocNLI dataset. Chapter 5 provides a detailed description of constructing the adversarial samples and coming up with an adversarial test set to estimate whether the DocNLI dataset suffers biases. Chapter 6 describes the results and detailed analysis of the results of chosen debiasing methods from the model perspective as well as from the data perspective. Finally, chapter 7 provides a conclusion of our work and outlook for future work.

2 Background and Related Works

In this chapter, we give an overview of the previous techniques and research that our work is based on. In Section 2.1, we describe the task of natural language inference at both sentence- and document-level, including the popular datasets used to train models for this task. In Section 2.2, we discuss the weakness of these datasets, in particular, different types of biases and their causes. Finally, in Section 2.3, we introduce different techniques that are developed over the years to mitigate the data biases from the datasets for the sentence-level NLI tasks.

2.1 Natural Language Inference

The development of the Natural Language Inference in the Natural Language domain can be traced back to the task of Recognising Textual Entailment (RTE) earlier introduced by Dagan et al. (2005) in PASCAL RTE challenge. Given two text fragments, namely, the premise (context), and the hypothesis, we want to predict the relationship between them in three-ways: the hypothesis (1) entails, (2) contradict, or (3) neutral to the premise. For example, given the premise "The actor is waiting for his sandwich to be served " and a hypothesis "A man is waiting for his food to be served ", a NLI system should be able to recognize that the premise is entailed by the hypothesis. However, recently, this task is widely known as Natural Language Inference as there are many critics argue that this task is not only to identify entailments. As the matter of fact, nowadays NLI models are required to not only recognize entailment but also to have reasoning ability and world knowledge to explain why a hypothesis contradicts a premise.

In recent years, there has been a surge of interest in document-level natural language understanding, which draws more attention to the task of NLI at the document level. The main reason for this is that many downstream NLP tasks can be cast as NLI problems, and having a universal well-performance can solve multiple tasks at once. A study by Mishra et al. (2021) has shown that document-level NLI is closely related to other tasks in NLP such as fact-checking and question-answering and we can indirectly improve those tasks by using a model that is trained on NLI datasets. For example, the task of fact-checking can be formulated into NLI where the evidence paragraph is the premise and the hypothesis is the piece of information we need to verify. However this poses a different problem to the traditional NLI systems, since the premise and hypothesis can span over multiple sentences and not limit to a single sentence, which challenge current NLI models to be able to work with longer text as well as to possess much more complex reasoning ability, as the difficulty and complexity increases with the length of the text. The major difference between the sentence-level and the document-level dataset is the length of the premise and the hypothesis. In a document-level dataset, the length of the premise and hypothesis can be varied from a single sentence to a paragraph or a whole document. This poses a great challenges to current NLI models considering the task of reasoning over long text is even challenging for humans.

2.1.1 Natural Language Dataset

Besides contextual understanding, we as humans intrinsically have a wide range of world knowledge and commonsense reasoning to be able to comprehend the nuances of languages. It is not surprising that developing NLI systems that reach the human-like ability still remains a great challenge despite significant advances in NLP. Subsequently, in order to accurately evaluate the ability of NLI models, many benchmark datasets have been created to help develop and evaluate NLI algorithms and models on their NLI performances.

Sentence-level NLI Dataset

The first sentence-level NLI dataset *RTE* is introduced by Dagan et al. (2005) is a combination of from a series of textual entailment challenges, namely RTE1, RTE2, RTE3 and RTE5. Examples are constructed based on news and Wikipedia text. Each of the dataset contains fewer than 1000 training examples. Even though it was introduced a long time ago, this dataset remains particularly challenging because many of its examples require broad world knowledge.

The next breakthough that supports the development of NLI task was *SNLI* dataset by Bowman et al. (2015), which is considerably bigger than the previous RTE dataset. The size of this dataset allows training deep learning models, and thus leads to a surge of works that employ much complex deep learning models afterwards. The reason behind the size of this dataset is that it was created by crowdsourced worker. In particular they are asked to write three sentences that entail, neutral and contradict to an images. Following the advent SNLI, other large-scale NLI dataset have been also created. For example, the *Multi-Genre NLI corpus* (Williams et al., 2018) is also a crowd-sourced collection of 433k sentence pairs, which is an extension of the SNLI corpus. However, it focuses on the improving the limitation of SNLI dataset, namely lack of variety of genres in both written and spoken English. Different from SNLI, MNLI covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation. Another dataset built on the SNLI is the *XNLI dataset* (Conneau et al., 2018) which serves as a benchmark for low-ressource language adaptation challenge in NLI. It contains examples in 15 different languages, each of the language set containing 5k training samples and 2.5k, which results in 112.5k annotated pairs.

Other than the line of dataset extended from the SNLI dataset, there is also a line of datasets that address special problems in NLI. For instance, *SICK* (Sentences Involving Compositional Knowledge) (Marelli et al., 2014) dataset consists of 9840 examples, which are rich in the lexical, syntactic and semantic phenomena. The goal of creating this dataset is to fill the gap with the previous dataset such as RTE and Frascal. It is constructed by randomly selecting a subset of sentence pairs from two sources - the 8k ImageFlickr dataset and the SemEval2012 STS MSR-Video Description dataset. In addition, there are two datasets that target at the Dialogue NLI. One example is the *Dialogue NLI Corpus* (Welleck et al., 2019) which consists of pairs of sentences generated using the PersonaChat dataset (Zhang et al., 2018a). Each human labeled triple is first associated to each persona sentence and then pairs of such triple; persona sentences are labeled as entailment, neutral or contradiction. The corpus consists of around 33k examples. Previous to Dialogue NLI, Zhang and Chai (2010) constructed a smaller dataset called *Conversation Entailment* consisting of 50 dialogues from the Switchboard corpus (Godfrey et al., 1992). 15 volunteer annotators read the dialogues and manually created hypotheses to obtain a total of 1096 entailment annotated examples.

Document-level NLI dataset

As mentioned earlier, document-level NLI has the potential to replace sentence-level NLI system, since it can serve as a universal Natural Language Understanding system to solve many tasks. Despite of this, there are only a few resources to facilitate the development of the NLI systems at the document level. It is caused by the novelty of the task as well as the huge work effort needed to construct such large-scale datasets, the number of document-level dataset is still limited. We will give a brief introduction of the available document-level dataset.

A popular approach to create document-level NLI dataset is to transforming large-scale dataset of various dataset into the NLI dataset. The first dataset that adopted this approach is the *QA-NLI* dataset (Demszky et al., 2018) consisting 500k NLI examples. The model learns a mapping from a question-answer pair into a declarative sentence, which allows us to convert question answering datasets into natural language inference datasets. More recently, DocNLI (Yin et al., 2021) surpasses QA-NLI by the size, as it comprises of >900 samples that is reformatted from datasets from a wide range of tasks than only Question Answering such as SQUAD for question answering, CNN/DailyMail, DUC, Curation for text summarization; and ANLI. Despite of its size that is practical for training deep learning models, a major limitation of this dataset is that it is automatically generated and thus lacks of post-editing. Eventually, the author highlights that DocNLI contains grammatical errors and also can have a high-level of overlap between the premise and hypothesis. The author also reveals a major setback of the raw version of DocNLI, as it contains hypothesis bias. In particular, they train a “hypothesis-only” baseline based on RoBERTa with the un-normalizing version of the data *raw*-DocNLI, this model performs better than the trivial baseline of 50%. As a result, Yin et al. (2021) performed an addition normalization step to make the “real” and “fake” hypothesis have the same number of instances of "entail " and "not-entail" class.

Although, the transformation approach enables the construction of large-scale document-level NLI, one major drawback is that these dataset does not provide any relationale that verify why the hypothesis entails or contradicts the premise. Considering the length of the premise and hypothesis, the entail or contradiction might come from a combination of different pieces of texts rather than a single piece of text. This makes it notoriously difficult to evaluate system which is trained or tested on these datasets, since there is no evidences to conclude that the models indeed makes the right predictions based on the correct rationale. To address this problem, Koreeda and Manning (2021) introduce *ContractNLI*, a fine-grained dataset with rationale. In ContractNLI, the premise is not the whole document as in QA-NLI and DocNLI but consists of multiple spans that support or refute the hypothesis. Moreover, instead of being crowdsourced or created automatically, this dataset was manually annotated by legal professionals. It was curated the annotation of 607 contracts over 13 different non-disclosure clauses. However, considering the size as well as the domain of ContractNLI, it is fairly impractical to use it for evaluating multi-genre document-level NLI systems.

Not only large-scale dataset that allows training deep neural network is important, it is equally crucial to have a benchmark dataset to assess the reasoning ability of document-level NLI models. Liu et al. (2021) introduce the benchmark dataset *ConTRoL* to address this problem. ConTRoL dataset contains 8,325 expert-designed "context hypothesis " pairs with gold labels that cover six reasoning skills e.g. coreferential reasoning, logical reasoning, temporal reasoning, information integration and analytical reasoning.

2.2 Bias in NLI datasets

Natural Language Inference is a challenging NLP task because it requires the model to be able to achieve human-level skills in understanding and reasoning. This requires the NLI model to learn the causal relationship between the premise and the hypothesis and the robust representation of this relationship. However, many works have recently revealed that many NLI models did not learn desired reasoning abilities. In particular, the models use the shortcut information or the spurious correlations between the input pairs to make the right prediction given incomplete information, thus drifting away from the fundamental problem of NLI. This can be problematic for evaluating the model performance with the in-distribution-test, with testset comes from the same dataset. Because the result of in-of-distribution tests is no longer an accurate measurement of model performance. This poses a huge risk if these models are deployed in effective environment, because we are unaware when the model will completely fail on the unseen dataset. Many reasons that can lead to model insufficiency: the first is the quality of the training dataset and the second can lie within the model architecture. Considering the data quality, a major reason is the presence of biases in the training dataset, which often refers to *spurious correlation*.

Formally formulated, a *spurious correlation* occurs when two variables are correlated but don't have a causal relationship. In NLP, a spurious correlation is a generic term that refers to dataset biases in which there is a correlation between certain biased features to a type of label. Consequently, the models might pick up spurious correlations in the training data. This explains why the model trained on a dataset containing spurious correlation can achieve good performance being tested on the in-distribution dataset but later suffers from a drastic drop in performance when tested on out-of-distribution datasets. This can be explained in detail as follows: in a normal setting where the biased feature does not present or the distribution of the label and the biased feature change, the model can't rely on the biased features, which leads to correct predictions earlier, thus leading to

Type	Name	Contained bias
Sentence-level	RTE (Dagan et al., 2005)	n/a
	SNLI (Bowman et al., 2015)	Hypothesis-only bias (Poliak et al., 2018); Gender, racial, religious, and aged-based stereotypes (Rudinger et al., 2017); Lexical heuristics (Glockner et al., 2018); Sentence length (Gururangan et al., 2018)
	MNLI (Williams et al., 2018)	Lexical heuristics, negation words (Naik et al., 2018); Sentence length (Gururangan et al., 2018)
	XNLI (Conneau et al., 2018)	n/a
	SICK (Marelli et al., 2014)	Negation, word overlap, and hypernym relations
	Dialogue NLI (Marelli et al., 2014)	n/a
	ANLI (Nie et al., 2020)	n/a
	Conversation Entailment (Zhang and Chai, 2010)	n/a
Document-level	DocNLI Yin et al. (2021)	n/a
	ContractNLI (Koreeda and Manning, 2021)	n/a

Table 2.1: Overview of biases in NLI datasets with the types of biases it contains and the authors discovered it.

a decrease in performance. In the following sections, we will discuss different types of biases and the main reason why they occur in the NLI dataset.

2.2.1 Types of biases in NLI datasets

A lot of work has been done to address the bias problem in popular sentence-level dataset. As mentioned above, biases and spurious correlation are often used interchangably. For better understanding, we would suggest to differentiate these biases into 5 different categories based on the biased features: *hypothesis-only bias*, *lexical biases*, *negation bias*, *representation bias*, and *social bias*. Table 2.1 gives an overview of the common NLI dataset and biases found in that dataset.

Hypothesis-only bias Poliak et al. (2018) reveal that the Stanford Natural Language

Inference dataset (SNLI; Bowman et al. (2015)) contains the most (or worst) hypothesis-only biases —their hypothesis-only model outperformed the majority baseline by roughly 100% (going from roughly 34% to 69%). In particular, they employ a simple fastText model trained only on SNLI hypotheses and find that this model can make the right prediction without having any access to the premise.

Negation heuristics Negation heuristics refers to the phenomena, in which the presence of the signal words makes the model geared toward the negative label Gururangan et al. (2018). Under the causal lens, negation heuristic is the confounding between the label and the use of the negation words and the lack of perfect balance in the probability of negation between entail and not-entailment examples. To illustrate this, we have an example: "they saw not one but three cats " entails "They saw three cats ". However, if the models is trained on a dataset such that the majority of the hypothesis containing the negation word "not " have contradict label, the model likely predicts that the two sentences are contradict based on the presence of the words "not "

Lexical biases Lexical biases implies to patterns in a dataset, which are created by repeatedly using similar or the same words in the hypotheses of the same class. Gururangan et al. (2018) provide many examples of lexical biases in the SNLI dataset. They observe that hypotheses that are labeled as a contradiction often contain some contradicting word such as 'sleeping' when the premise is about an activity. In addition, they observe that modifiers, superlatives or purpose clauses in hypotheses often affliliate with the neutral class. They also observe that the hypotheses contain a lot of generic words and approximations for the entailment class.

Word overlap heuristics Naik et al. (2018) inspect many cases where the NLI model fails and find that the first main reasons for the model failure at inference time is due to the shallow lexical learning without having the semantic understanding of the sentence, in which they rely too much on word-overlap. Gururangan et al. (2018) found in SNLI the entailment class hypotheses were also often shorter and had a lot of overlap with their premise. McCoy et al. (2019) also reveal that the high overlap between the premise and the hypothesis can be the cause for the biased prediction of the model toward entailment. For example, if a dataset contains a lot of samples with high word overlap between the premise and the hypothesis, in other words, two sentences with the same word but different word order are different in meaning "*The dog chased the cat*" and "*The cat chased the dog*", the model will likely identify a pair of premise and hypothesis to be entailed if they have high word-overlap rate and contradict if their word-overlap is low without considering the semantic relationship between the two.

Social bias Rudinger et al. (2017) show that the elicited hypotheses in SNLI dataset introduced substantial gender stereotypes as well as varying degrees of racial, religious, and age-based stereotypes. Sharma et al suggest that three models (BERT, RoBERTa, BART) trained on MNLI and SNLI datasets are significantly prone to gender-induced prediction errors. They suggest to augment the training dataset to ensure a gender-balanced distribution proves to be effective in reducing bias.

2.2.2 Cause of biases

The main cause accounted for the biases in the dataset is the quality of the annotators. Because MNLI and SNLI are created through crowdsourcing, the annotators, non-language expert, cannot use a wide range of vocabulary and poor choice of expressions to construct the hypotheses, introducing the lexical bias. Moreover, crowd workers often negate the premise sentence by adding simple negation words to save time, thus induce the correlation between negation words (e.g., “not”) and the contradiction label. The second cause of biases comes from the lack of data curation and quality control in the data generation process. It is important to improve linguistic diversity in NLI datasets, which cannot be achieved without data quality consideration. Because performing quality check are extremely resource intensive, given the amount of data required, this is often neglected despite of its importance.

2.3 Debiasing techniques on NLI datasets

Having defined what is bias in NLI dataset, the following section will describe the different techniques to mitigate the data biases problem in prior works. We distinguish between two de-biasing techniques: *data-centric* and *model-centric* methods. While data-centric methods focus on improving the quality of the training dataset to prevent the model from exploiting spurious correlation, model-centric methods strive for enhancement of NLI models against prevalent biases in the dataset without dealing with the data. In this section, we will give an overview of existing methods used to counter and discuss their advantages and disadvantages in detail. Table 2.2 shows an overview of the existing de-biasing methods on NLI datasets and their corresponding types of biases as in Section 2.2.1 which they can mitigate.

Class	**Technique**	**Type**	**Datasets**
Data-centric	AFlite (Le Bras et al., 2020)	Data Filtering	SNLI
	Z-filtering (Wu et al., 2022)	Data Filtering	SNLI, MNLI
	Generation and filtering (Wu et al., 2022)	Data Augmentation + Filtering	SNLI, MNLI
	Data cartography (Swayamdipta et al., 2020)	Data Augmentation	SNLI, MNLI
	(Kaushik et al., 2020)	Data Augmentation	SNLI
	(Zhou and Bansal, 2020)	Data Augmentation	MNLI
	(Moosavi et al., 2020)	Data Augmentation	MNLI
Model-centric	(Clark et al., 2019)	Ensemble learning	MNLI
	(Sanh et al., 2021)	Ensemble learning	MNLI
	DRIFT (He et al., 2019)	Ensemble learning	SNLI, MNLI
	(Clark et al., 2020)	Ensemble learning	MNLI
	(Utama et al., 2020a)	Ensemble learning	MNLI
	(Utama et al., 2020b)	Debiased training objectives	MNLI
	(Karimi Mahabadi et al., 2020)	Debiased training objectives	MNLI
	(Utama et al., 2021)	Prompt-based Finetuning	SNLI,MNLI
	(Belinkov et al., 2019)	Adversarial learning	MNLI, SICK
	(Joshi et al., 2022)	Causal Inference	

Table 2.2: Overview of debiasing methods on NLI datasets with their types of debiasing strategy and tested sentence-level datasets

2.3.1 Data-centric debiasing methods

One interesting perspective to view the data-centric debiasing methods is through the causal lense. In causal inference, variables such as the existence of negation words, affecting both the input X and the answer Y is called a confounder Z, which disturbs a true causal relationship between X and Y. The causal intervention intentionally diminish the relation between X and Z. Many data-centric debiasing techniques have been proposed all try to tackle the bias problem by de-confounding. These existing methods is best classified into (i) filtering and (ii) data augmentation, and (iii) hybrid. An intuitive strategy to mitigate bias in the dataset is to directly remove the biased instance to prevent the model from learning from biased features, this approach is often referred to as *filtering*. The second approach is to generate more data to transform a skewed data distribution so that the distribution of the biased feature become uniform and cannot affect the model learning anymore, this refers to *data generation* approach. We can combine both the filtering and the generation approach to create a the third approach, the *hybrid* approach.

Filtering
Le Bras et al. (2020) studied the *AFlite* algorithm from the earlier work by Sakaguchi et al. (2019) to create the Winogrande dataset and put it on the test with different NLI datasets. The AFlite algorithm aims to prevent the model from learning bias from the rich head of the data distribution while preserving the complexity of the tail. Their hypothesis is that the training instance that contains spurious correlation will have generally high confidence for prediction score and thus we can use these characteristics to identify and remove them without prior knowledge about the biases. Swayamdipta et al. (2020) propose to group training instances into hard- and easy-to-learn subsets based on the model's confidence in the true class, and the variability of this confidence across epochs. They hypothesize that hard-to-learn instances are instances that have low confidence and easy-to-learn would have high-confidence. They prove that by focusing on the hard-to-learn instances the model can learn a better representation of the data and thus performs better on the out-of-distribution set. In another work, Wu et al. (2022) utilizes the z-statistics of each instance to filter out an instance that contains the biased feature, which is known as *z-filtering* algorithm. The main idea of this approach is to identify and remove "easy" instances and train the model on the "hard" subset of the training data. In statistical machine learning, this is also known as Down-sampling. However, we should note that this approach can also remove useful linguistic phenomena that the model should learn (Liu et al., 2020).

Data Augmentation
In contrast to the filtering method, data augmentation techniques do not explicitly remove biased but retain them in the dataset and attempt to balance out its effects. With spurious data biases, the common approach is label balancing to transform the data distribution between all features and label classes to be uniform. This approach can be realized by first identifying the biased feature of the training instances and then adding samples of the opposite class of the biased feature to remove the correlation between the label and the biased features. For example, Zhou and Bansal (2020) suggest creating data balance between classes by repeating training data to remove lexical bias. Kaushik et al. (2020) also augmented the data with counterfactual examples to weaken the effect of spurious correlations on the models. Moosavi et al. (2020) found that adding more training data by augmenting sentences with their corresponding predicate-argument structures robustified NLI models against different types of biases. Similar data augmentation via syntactic transformation has also suggested by Min et al. (2020) to diversify NLI datasets.

Hybrid methods
Besides the two main approaches mentioned above, Wu et al. (2022) introduce hybrid methods that combine both data generation and data filtering to combat the data bias problem by using a data generator to create more sample and then use the z-filtering method to extract the high-quality data sample.

2.3.2 Models-centric debiasing methods

Along with data-centric methods, there exists many approaches to robustify neural networks against bias in the dataset, refers as model-centric methods. As seen in Table 2.2, the model-centric methods are more popular than data-centric methods. It might be due to one advantage of model-centric methods that there is no need to know the specific bias type in advance.

Ensemble learning
The most effective line of work so far has been ensemble based methods. This is also the most popular method to combat against bias problems in the NLI dataset. Ensemble-based methods provide a way to combine different learners with different capabilities such that the weak model will capture the biases of the dataset and the main model can learn from the weak model to later avoid these biases. Sanh et al. (2021) suggest by combining a weak learner and a strong learner, the more robust model can learn from the error of the weak learner. The final loss is a linear combination of the cross-entropy loss and the product-of-expert loss, also refers as a multi-loss objective. Clark et al. (2019) train a naive

model that makes predictions exclusively based on dataset biases, and (2) train a robust model as part of an ensemble with the naive one in order to encourage it to focus on other patterns in the data that are more likely to generalize. This requires prior knowledge about the bias. He et al. (2019) train a debiased model that fits to the residual of the biased model, focusing on examples that cannot be well predicted by biased features only. Another example, Stacey et al. (2020) mitigates hypothesis-only bias in the SNLI dataset by using an ensemble of the different adversarial classifiers. Clark et al. (2020) trains a lower capacity model in an ensemble with a higher capacity model.

Debiased training objectives
Along with ensemble learning, there is also a line of works that employ a training objective concretely to mitigate the bias problem in NLI dataset. Different from above mentioned debiasing techniques such as ensemble learning or data-centric techniques, which was original developed for other tasks, debiasing techniques with debiased training objectives are developed explicitly to tackle this bias problem in NLI dataset. A wide range of debiasing losses are based on the proposition that the level of bias of each training instances can be estimated through a biased model's predictions. They use this insight to remove the biased instances. Among the most common is product-of-expert (Schuster et al., 2019). Another example is learned-mixin (Clark et al. (2019), Utama et al. (2020b). Recent work introduces various learning algorithms to avoid adopting heuristics including by re-weighting (He et al. (2019); Karimi Mahabadi et al. (2020); Clark et al. (2020)) on the training instsances which exihit certain biases.

Confidence regulation
Addressing the limitation of reweight instances that the in-distribution performance decrease while the out-distribution increase, Utama et al. (2020a) regularize the confidence on the training instances which might be biased. Their methods accomplish both goals to robustify the models towards unseen instances without degrading the in-distribution performance.

Complementary to the mainstream lines of works, alternative methods have tackled this issue in various ways. Belinkov et al. (2019) make use of an adversarial training technique to remove hypothesis-only bias from the input representation using. Yaghoobzadeh et al. (2021) employ a two-stage training paragidm, first they train their models the full training dataset, after that they train the model again on minority examples, also called forgettable examples. They define an training instance as example forgettable if during training it is either properly classified at some point and mis-classified later, or if it is never properly classified. Utama et al. (2021) mitigate the heuristic learning in Prompt-based PLM Finetuning by adapting a regularized prompt-based fine-tuning based on the Elastic

Weight Consolidation (EWC) method. Meissner et al. (2022) assume that the model exploits the bias in the dataset and later on bias is caused by a certain subset of weights in the model. They reframe the debiasing process as a mask search, in which they put mask in individual weight. Afterwards, they search for a pruning mask that cancels out the weights that cause biased outputs, producing the desired effect without altering the original model.

3 Baseline reproduction experiments

The model and the dataset presented in this work are built upon the based line models and the dataset in DocNLI. In this chapter, we focus on replicate the results from the paper which introduce DocNLI (Yin et al., 2021). First, we give a brief introduction of the DocNLI dataset and the baseline models. Then, we start with reproducing the results in the paper Yin et al. (2021) to verify them. These results serve as the baseline for all of our further experiments in this thesis.

3.1 Dataset

In this section, we present a few key statistics of the DocNLI dataset[1]. Table 3.1 shows the sizes train/dev/test in two classes " and -entail" in the DocNLI dataset. As we can see, the number of samples in two classes are balanced for the training set, while the numbers of -entail" is roughly 7-times larger than the numbers of "entail" in the development and the test set. We consider this imbalance as a flaw of the DocNLI dataset. Because the testset is heavily skewed, it can be hard to accurately interpret the performance of the model based on this testset. The class imbalance also applies for the development set and thus makes it difficult to validate the best models.

Figure 3.1 illustrates the length distribution by tokens of the premises and hypotheses in DocNLI, it can be clearly observed that the length of the premise and hypothesis in DocNLI exceeds the length of sentence-level NLI dataset. The majority of the premises of DocNLI has 1-150 words, which spans around 7-10 sentences, as a sentence typically has 15–20 words. The majority of the hypothesis in DocNLI has 50-100 words, which is equivalent to 3-6 sentence. As a result, a premise is almost two times longer than a hypothesis for the majority of the instances.

[1] https://drive.google.com/file/d/16TZBTZcb9laNKxIvgbs5nOBgq3MhND5s/view?usp=sharing

	train	dev	test
entail	466,653	28,890	33,128
not_entail	475,661	205,368	233,958
sum	942,314	234,258	267,086

Table 3.1: Data sizes of DocNLI

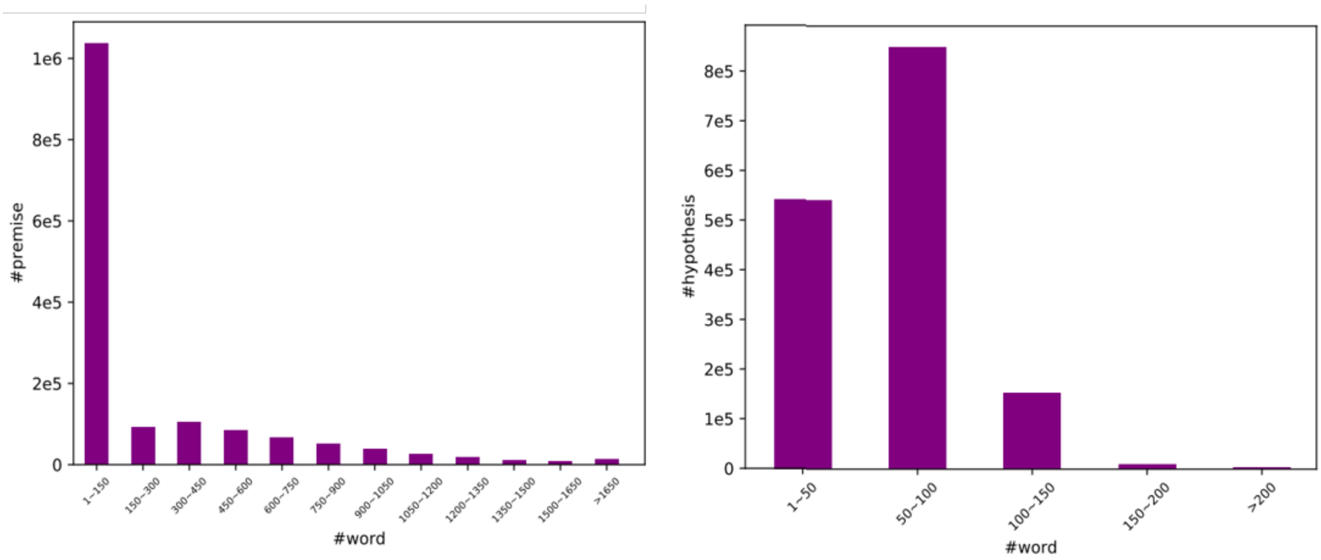


Figure 3.1: Distribution of premise and hypothesis length in DocNLI

3.2 NLI Models

Yin et al. (2021) employ two state-of-the-art pre-trained language models to create the baseline model on DocNLI, namely RoBERTa (Liu et al., 2019) and Longformer (Beltagy et al., 2020). To be more concrete, the authors use the RoBERTa-large variant, which can handle maximal 512 tokens; and Longformer-base, which can handle up to 4096 tokens, however due to memory constraint they limit the maximal tokens length to be 1300 tokens. Longformer is a transformer-based models that is developed to specially handle with long text, such that it has sliding window attention with global attention to replace the self-attention mechanism in pre-trained Transformers. Table 3.2 provides keys

	RoBERTa-large	Longformer-base
Max token length	512	4096 1300 (ours)
Details of the model	12-layer, 768 hidden, 12 heads, 125M parameters	12-layer, 768-hidden, 12-heads, ~149 parameters
Shortcut name	roberta-large	allenai/longformer-base-4096
Batch size	1	4
Learning rate	5e-6	1e-6
Epoch	5	5

Table 3.2: Model configurations of DocNLI baseline models

information of the two baseline models. Based on the implementations provided by the authors, the NLI models was built as follows: Given a premise p and a hypothesis h, a new sequence [CLS]$+\ p\ +$ [SEP] $+\ h\ +$ [SEP] was created by concatenating the premise and the hypothesis, where [CLS] and [SEP] are the classification token and separator token respectively. This sequence is feed in the models for pretrained model encoding, after that the last layer's hidden representation from the [CLS] token is is passed through another multi-layer perceptron (MLP) to get the final NLI classification labels.

3.3 Reproduction results

We conduct two different reproduction experiments to test the in-distributions performance of the two baseline models, we use the model implementation by Yin et al. (2021) with the code and the data provided in the GitHub[2] repository. The author only provides the performances of the NLI model which is based on the RoBERTa-large pretrained LM, there is no evaluations of the NLI models based on the Longformer-base pretrained LM. They argue that the Longformer-based model performance is much lower than RoBERTa-based model, thus Longformer-based model is no longer considered for any further experiments in their paper. However, we would like to take Longformer-large into consideration for the sake of comparison with RoBERTa. For this reason, we retrained two baseline models with both pre-trained LMs RoBERTa-large[3] and Longformer-base[4] on the training data with the configurations provided by the authors in the paper as in Table 3.2. We report our results

[2] https://github.com/salesforce/DocNLI

[3] https://huggingface.co/roberta-large

[4] https://huggingface.co/allenai/longformer-base-4096

	dev	dev (Ours)	test	test (Ours)
Hypothesis-only	0.2189	0.4050	0.2202	0.3908
RoBERTa-large	0.6305	0.6340	0.6120	0.6120
Longformer	0.4618	0.4667	0.4442	0.4456

Table 3.3: Comparisons of the baseline reported by Yin et al. (2021) and our reproduction on the F1 score on the entailment class

on the F1-scores of the baseline models in Table 3.3. We notice no big difference between our results and the results provided by the authors on the F1 score for the entailment class as seen in Table 3.3.

Nevertheless, we doubt that the F1 score on the entailment class can be a sufficient metric to evaluate the models in further steps. Because of this reason, to make a better overview of the model capability, we additionally report the F1 for the non-entailment class as well other metrics for a better overview of the two model capability as shown in Table 3.4. In this evaluation, we can observe that the F1 score on entail class has dropped by 0.03, we do not consider this as abnormal since the values can differ in each running round.

When we compare the performance between two label classes, we can see that both of the models have a better performance on the not-entail class than the entail class. Taken all the metrics of the entail class into consideration, we can observe that the precision, recall and F1 score values approximately equal to a random guess at 0.50. It can be concluded that models trained on DocNLI wrongly predict many entail samples as non-entail. We speculate that the insufficient capability of the pre-trained LMs accounts for the model bad performance. Here, we would exclude the data imbalance as a possible cause for the model biased prediction towards the not-entail class. Because, the class labels in the training set are balanced (see Table 3.1). This finding is crucial and we will examine it more in the following experiments in our works.

	RoBERTa			**Longformer**			Support
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	
Entail	0.48	0.74	0.58	0.62	0.35	0.45	33128
Not-entail	0.96	0.88	0.92	0.89	0.64	0.74	233958
Accuracy			**0.87**			**0.69**	267086
Macro Acc.	0.72	0.81	0.75	0.76	0.50	0.60	267086
Weighted Avg.	0.90	0.87	0.88	0.86	0.60	0.71	267086

Table 3.4: Our reproduction of Longformer and RoBERTa howing the main classification metrics precision, recall, F1, and accuracy

4 Bias Discovery

In the previous chapter, we reproduce the baseline by the author of the paper DocNLI(Yin et al., 2021). In this chapter, we propose a preliminary approach for bias discovery within the DocNLI dataset. More specifically, we conduct different experiments to determine whether the DocNLI dataset indeed contains certain biases. We perform two types of assessments: direct assessment with statistical tests to measure the hypothesis-only bias and indirect assessment, in which we compare the performance of the model trained on the DocNLI dataset on an in-distribution dataset (test set of the DocNLI) and the out-distribution dataset, a diagnostic testset to measure the lexical and syntactical biases. Section 4.1 presents our results for the statistical tests and Section 4.2 reports our results of the diagnostic tests.

4.1 Statistical tests

As mentioned earlier in chapter 2, a large source of biases can be unconsciously added to the dataset if the data generation is not controlled carefully. In this section, we aim to demonstrate the presence of hypthesis-only biases in the DocNLI dataset. In the following sections, we will discuss the statistical methods we used to measure the heuristics in DocNLI and ContractNLI dataset and report the results of these tests.

4.1.1 Methods

In this section, we describe the methodology to conduct the experiment to quantify the spurious correlations between the labels and the sample as the first step to address them. In particular, we measure the correlation between various words (1,2,3,4-grams) in the hypotheses of DocNLI training set and labels, where a n-gram tuple is denoted as w_i and the specific labels as l_j of two classes "entail" and "not-entail". In NLP, n-grams is a set of co-occurring words within a given window. Given a sentence "The cow jumps over the moon". If N=2 (known as bigrams), then the 2-grams would be: ["the cow", "cow jumps", "jumps over", "over the", "the moon"]. If N=3 (known as trigrams), the 3-grams would be: ["the cow jumps", "cow jumps over", "jumps over the", "over the moon"]. We use two simple metrics to value the correlations between a specific n-grams and a specific labels: conditional probability $P(l_j \mid w_i)$ and point-wise mutual information $PMI(w_i, l_j)$. Our goal is to find the n-grams that has the potential to be the predictive bias in the DocNLI dataset. Our workflow to calculate the PMI and conditional probability can be described as follows: First, we convert the text into ngrams of size [1,2,3,4]. Although more sophisticated feature can be considered such as skip-grams, we restrain from using it because it would be computationally too expensive. Second, we calculate the number of times a n-gram occur with a specific labels, this is a raw count. In the last step we calculate the PMI between a n-gram with a specific labels.

PMI (Point wise mutual information) Using PMI to measure statistical irregularity is not new in NLP (Gururangan et al., 2018). PMI is a non-parametric measure of statistical association derived from information theory. It computes the log probability of co-occurrence scaled by the product of the single probability occurrence that ranks the statistical dependence between two random variable. Given the probability $P \in \Delta(\Omega)$ and that events $A, B \subseteq \Omega$. The general formula is defined as follows:

$$PMI(A,B) = \frac{\log_2 P(A,B)}{P(A)P(B)} = \log_2(\frac{P(A \mid B)}{A}) \tag{4.1}$$

When A and B are perfectly correlated,

$$P(A \mid B) = P(B \mid A) = 1 \tag{4.2}$$

we have the following:

$$PMI(A,B) = \frac{1}{P(A)} \tag{4.3}$$

Therefore, less frequent n-gram will have higher PMI score than frequent n-gram, even if both are perfectly correlated with the label. Furthermore, as we can see with this general formular, the PMI values are hard to compare. Because in case A and B are perfectly associated $P(A \mid B) = P(B \mid A)$ the PMI maximizes yielding the following bound $-\infty \leq pmi(A,B) \leq min[-log_2 P(A), -log_1 P(B)]$. This bound are in different scale such that the ranking of pmi values become impossible. To overcome this limitation of the general PMI, we use the *normalized* PMI so that the comparison of the PMI values will be easier. The *normalized* PMI can be described as follows:

$$npmi(A,B) = \frac{pmi}{-log_2(p(A,B))} = \frac{log_2[P(A)P(B)]}{log_2 P(A,B)} - 1 \tag{4.4}$$

This normalized lets us have the following interpretation about the npmi(A,B):

1. if there is no co-occurrences between A and B, thus $log_2 P(A,B) - \infty$, then $nmpi = -1$
2. else if the two events A and B randomly co-occurrence, thus $log_2 P(A,B) = log_2[P(A)P(B)]$, so $nPMI = 0$
3. only when A and B are perfectly co-occurrence, $log_2 P(A,B) = log_2 P(A) = log_2 P(B)$, then $npmi = 1$

On the contrary to pmi, the normalized pmi is bound between $[-1,+1]$, which allows us to compare n-grams with very different frequency. In our case, we would adapt the PMI to measure the association between a n-gram and a specific label and define the PMI as follows:

$$nPMI(w_i, l_j) = \frac{\log_2[P(w_i)P(l_j)]}{log_2 P(w_i, l_j)} - 1 \tag{4.5}$$

$$P(w_i) = \frac{\text{count}(w_i)}{\text{Total occurrences of} w} \tag{4.6}$$

$$P(l_j) = \frac{\text{count}(l_j)}{\text{Total number of samples}} \tag{4.7}$$

$P(w_i)$ is the probability of the n-gram w_i occurring, which is the ratio between the number of occurrence of a unique n-gram w_i and the total numbers of n-grams of the documents. $P(l_j)$ is the probability of the a specific label class. $P(w_i, l_j)$ is the ratio between the number of times a specific n-gram w_i associates with the specific label l_j

Conditional probability Followed Poliak et al. (2018) and Gardner et al. (2021), we employ conditional probability to find out highly label-specific words. Assuming probability $P \in \Delta(\Omega)$ and events $A, B \subseteq \Omega$, the conditional probability of A given B, written as $P(A)$, gives the probability of A on the assumption that B is true. The formula for conditional probability can be written as follows:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \tag{4.8}$$

which we can also rewrite as:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \tag{4.9}$$

In our case, we would define the two events A and B as the label l_j and the n-grams w_i. Then, we can determine the conditional probability of a label l_j given the n-grams w_i as follows:

$$P(l_j \mid w_j) = \frac{count(w_j, l_j)}{count(w_j)} \tag{4.10}$$

$count(w_i, l_j)$ is the number of times of a specific n-gram appearing along with a specific label and $count(w_i)$ is the number of times a specific n-gram appear in the whole hypotheses of the dataset. Suppose $P(l_j|w_i)$ is highly skewed across labels, it can potentially give rise for a predictive bias. As a result, a word w serves as "give-away" cue for the model, which allows it to make right predictions without considering the context. In DocNLI, there is two class labels, therefore, if $P(l_j \mid w_i) \leq 0.5$ the label l_j does not correlate to word w_i and vice versa. The bigger the $p(l_j \mid w_i)$ values, the more likely for it to be a "give-away" word. In a dataset that is free from hypothesis-only bias, we would expect the marginal distribution over labels given any single feature is uniform. Concisely formulated, we assume a feature w_i and an output value l_j, where $x \in$ n-grams in hypothesis, the hypotheses are considered to be unbiased, when $P(l_j|w_i) \leqq 0.5, \forall i$.

4.1.2 Results

Contradicted to our earlier belief, DocNLI does not suffer from hypothesis-only bias based from the results of our two statistical tests. When we observe the conditional probability score of all $[2, 3, 4]$-grams given the label in DocNLI, there is no n-grams with $p(l_j \mid w_i) \geq 0.5$. For the PMI, the maximal normalized value $PMI(w, l)$ over all n-grams is 0.3, which indicates that there is no n-grams in the hypothesis highly affiliated with any label. Nevertheless, we find a possible reason that we cannot find hypothesis-only bias in DocNLI. Yin et al. (2021) perform a balance step as a counter-measurement to their findings about the hypothesis-only bias, which turn out to be successful. In their original raw-DocNLI dataset, the hypotheses of the not-entail class are fake summary created by a generative model CTRL. These fake summary might have very different distribution than the real hypotheses and more important they are only affiliated with the not-entail class label. Thus, there might be some linguistic properties of those fake summary that allows the model to make the right prediction without even consider the premise. The authors have added more instances such that the fake summaries or the hypotheses do not only co-occurrence with the not-entail class but also appear with the entail class. This treatment breaks the association of the class not-entail with the fake hypotheses in the original raw-DocNLI dataset.

4.2 Diagnostic tests

In this section, we continue using different techniques to diagnose the bias in the DocNLI dataset regardless of the results of the statistical tests. Although we could not find any evidence that DocNLI indeed contains any hypothesis-only bias, there are many other types of biases that sentence-level NLI models are proved to adopt from the training set. Being aware that these potential pitfalls might also apply for document-level dataset, we would like to investigate the DocNLI for further types of biases, particularly lexical and syntactical heuristics.

4.2.1 Methodology

In contrast to hypothesis-only bias, lexical and syntactical biases are harder to be detected, as they might not comprise of a individual feature such as negation word but through a complex interaction between features. Due to this reason, it would be impossible to calculate the statistics from all combinations of words to find the biased features. In order to overcome this problem, we opt for a way to evaluate the dataset without having to measure the exact biases in a dataset, which would be to test the model trained on the targeted dataset on a diagnostic testset. The goal of these diagnostic testset here is to reveal the weakness of the models w.r.t certain heuristics. These diagnostic dataset typically contains "tricky" samples with certain types of heuristics. If the models performance on the adversarial testset is substaintially lower than its performance on it native testset, we hypothesize that the dataset can contain these types of heuristics as in the adversarial dataset. The NLI model can adopt these types of heuristics in its representation layer, uses these heuristics to make prediction and thus will fail when it encounter the "tricky" case in the adversarial testset. Many prior works have also employed this diagnostic testset to reveal the biases in sentence-level NLI dataset (Glockner et al. (2018), Naik et al. (2018), McCoy et al. (2019), Gururangan et al. (2018), Nie et al. (2020)).

We perform two types of diagnostics test: (1) directly test on the diagnostic testset (HANS and ANLI) and (2) with simple augmentation techniques on our test set with HANS and ANLI. We then measure the accuracy of the baseline model RoBERTa-large trained on the original training set in these two tests. In the first experiments, we test the DocNLI model on the original HANS dataset. In the second experiments, we create a augmented dataset by appending the samples from HANS dataset to the DocNLI dataset. We evaluate each model on both adversarial sets to examine whether the models trained on DocNLI training sets is robust towards both heuristics being tested for.

4.2.2 Diagnostic Dataset

We will give a brief description of the adversarial dataset, which we use as basis for our diagnostic tests, namely HANS (McCoy et al., 2019) and ANLI (Nie et al., 2020). They are the most common benchmark datasets to assess on the shortcomings of current NLI models, which is widely used in many prior works. Each dataset contains three subsets train, development and test. For testing purpose, we only choose the test subset for the experiments below. Furthermore, because DocNLI baselines are trained on DocNLI dataset with only two labels without the "neutral" class, however both ANLI and HANS are three-way NLI with "neutral". For this reason, we only consider the instances of the two class "entail" and "not-entail" in the ANLI and HANS testset.

ANLI

The Adversarial Natural Language Inference (ANLI) Nie et al. (2020) is a large scale NLI benchmark dataset, which is collected via an iterative, adversarial human-in-the-loop procedure. The focus of ANLI dataset is on the non-trivial NLI relationship between the premise and hypothesis. In particular, ANLI contains six types of inference: Numerical Quantitative (i.e., reasoning about cardinal and ordinal numbers, inferring dates and ages

Premise	Hypothesis	Label	Types of reasonings
Roberto Javier Mora Garcıa (c. 1962 – 16 March 2004) was a Mexican journalist and editorial director of “El Manana”, a newspaper ∼ based in Nuevo Laredo, Tamaulipas, Mexico. He worked for a number of media outlets in Mexico, including the “El Norte” and “El Diario de Monterrey”, prior to his assassination.	Another individual laid waste to Roberto Javier Mora Garcia.	Entail	Lexical (assassination, laid waste)
Sergei Mikhailovich Grinkov (Russian: , February 4, 1967 — November 20, 1995) was a Russian pair skater. Together with partner and wife Ekaterina Gordeeva, he was the 1988 and 1994 Olympic Champion and a four-time World Champion."	Sergei Mikhailovich Grinkov became the 1988 Olympic Champion with his partner while his wife cheered from the audience	Not-entail	Reasoning (Facts), Reference (Coreference)

Table 4.1: Examples of premises and hypotheses from the ANLI (Nie et al., 2020) dataset.

from numbers, etc.), Reference Names (coreferences between pronouns and forms of proper names, knowing facts about name gender, etc.), Standard Inferences (conjunctions, negations, cause-and-effect, comparatives and superlatives etc.), Lexical Inference (inferences made possible by lexical information about synonyms, antonyms, etc.), Tricky Inferences (wordplay, linguistic strategies such as syntactic transformations/reorderings, or inferring writer intentions from contexts), and reasoning from outside knowledge or additional facts (e.g., “You can’t reach the sea directly from Rwanda”). Table 4.1 shows some examples of the test instances in ANLI dataset.

HANS

HANS (Heuristics Analysis for NLI System) focuses on heuristics that are based on syntactic properties, consisting of more than 30k samples. It contains many examples designed to reveal syntactical heuristics in the models, in such a way that models suffering from these heuristics will perform poorly on HANS. The syntactical heuristics in HANS belong to three groups: lexical overlap, subsequence heuristic, and the constituent heuristics. Lexical overlap heuristic arises when a premise entails all hypothesis constructed from words

Premise	Hypothesis	Label	Types of heuristics
The professor who introduced the doctors recognized the secretaries	The doctors recognized the secretaries.	Not-Entail	Subsequence
Certainly the senators recognized the actor.	The senators recognized the actor	Entail	Constituent
The professors advised the judge.	The judge advised the professors.	Not-entail	Lexical overlap

Table 4.2: Examples of premises and hypotheses from the HANS McCoy et al. (2019) dataset

in the premise (word order is ignore). Subsequence refers to a phenomena when the premise entails all of its contiguous subsequences (relation between phrases is ignored). Constituent heuristic happens when the premise is always entailed all complete subtrees in its parse tree. To illustrate, we presents examples from the HANS dataset in Table 4.2.

4.2.3 Results

Table 4.3 shows that the RoBERTa-large model, which was trained on the DocNLI data exhibits a substantial performance drop on the HANS diagnostics dataset on both of the experiments. The F1 score by label and the overall accuracy all drop below the value of the corresponding baseline on DocNLI testset (in orange). In general, the results of our diagnostic tests show that the DocNLI dataset might potentially suffer from the bias problems which are reflected in the HANS and the ANLI dataset such as word-overlaping heuristics or syntactical heurisitics. Our hypothesis here would be that the NLI models might adopt these heuristics is that their input representations rely on these heuristics to make a correct prediction in the DocNLI, and thus fail in the diagnostic test set.

To follow up with this finding, we perform extra experiments, in which we measure the accuracy of the baseline model RoBERTa-large on HANS dataset by subcases. This kind of experiments would not be possible with ANLI, because the test instances in ANLI is not labeled with the type of heuristics it contains. However, the HANS subscases alone already sheds some light on which types of heuristics the model most suffers from. More specifically, we test the baseline model RoBERTa-large on two testset, the first testset is the original HANS test set by subcase. The second testset used is constructed from by appending a random premise and hypothesis pair from HANS to DocNLI testset. Given

$\{p_{HANS}, h_{HANS}\}$ as the premise-hypothesis pair from HANS, and $\{p_{DocNLI}, h_{DocNLI}\}$ as the premise-hypothesis pair from DocNLI testset, we have the appended testset $\{(p_{DocNLI} + p_{HANS}), (h_{DocNLI} + h_{HANS})\}$. The goal of the appended testset is to minimize the difference between the test instances and the training instances, such that we have a more accurate evaluation of the model's ability. It would be difficult for the model when it is trained on a document-level dataset (DocNLI train set) but is tested on a totally different testset, which is at sentence-level (HANS and ANLI).

Table 4.4 reports the accuracy by subcase in the HANS test dataset ranked by the decreased values. It is clear to observe that RoBERTa baseline model does not suffer from all types of heuristics, because the accuracy of the subcases on the upper-half of the table are even higher than the accuracy of the RoBERTa baseline (0.87). A second interesting observation is that RoBERTa baseline model performs well in predicting the *not-entail* relationship better than the *entail* relationship. We can see that from the subcases where it performs better (the upper half), 13 out of 16 subcases are *not-entail* subcases. In contrast, 2 out of 14 subcases in the lower half are *not-entail* subcases. Moreover, the performance seems to be only affected from the labels and not from the type of heuristics, because the three type of heuristics (lexical, subsequence and consequence) are distributed equally in the upper-half and lower-half of the results.

Combining this observation and the results we observed from the reproduction the baseline in Chapter 3, we make an assumption about the DocNLI dataset as follows: the model capability is limited to recognizing *not-entail* than *entail* relationship because it relies too much on the superficial cues to predict the *not-entail* class correctly most of the time. It particularly only focuses on the difference between the premise and the hypothesis or in other words the portion of the hypothesis that does not exist in the premise. This can be the result from a poor linguistics phenomena in the DocNLI dataset, because the a large portion of the *not-entail* hypotheses are created from *entail* hypothesis by adding random noun phrases into to the original hypothesis through word replacement and entity replacement. In this way, any hypothesis that contains new words from the premise would be predicted as *not-entail*.

4.3 Discussion

In this chapter, we perform the preliminary study on the DocNLI dataset to reveal its potential biases: statistical tests and diagnostic tests. Having demonstrated by these tests, we can draw two important conclusions. First, DocNLI dataset does not suffer from the

subcases		precsion	recall	f1-score	accuracy
entail	DocNLI	0.48	0.74	0.58	
	ANLI	0.68 (↑ .20)	0.46 (↓ .28)	0.54 (↓ .04)	
	HANS	0.71 (↑ .23)	0.88 (↑ .14)	0.42 (↓ .16)	
not-entail	DocNLI	0.96	0.78	0.92	
	ANLI	0.59 (↓ .37)	0.78	0.67 (↓ .25)	
	HANS	0.56 (↓ .40)	0.88 (↑ .10)	0.68 (↓ .24)	
all (entail + not-entail)	DocNLI				0.87
	ANLI				0.62 (↓ .25)
	HANS				0.59 (↓ .28)

Table 4.3: Result of testing the baseline model RoBERTa-large on different testset (1) DocNLI testset, (2) ANLI, and (3) HANS with the difference between the out-of-distribution accuracy (2,3) from the in-of-distribution accuracy (1)

hypothesis-only biases. Second, DocNLI dataset, however, exhibits bias to certain types of heuristics, particular constituent, word-overlap and subsquence heuristics. A possible explanation for this might be that the baseline models trained on DocNLI have inherited those biases from its source dataset, which are reformatted into DocNLI.

Although, our tests provide important observations to identify the biases in DocNLI dataset, we acknowledge that the reliability of our diagnostic tests are still hindered by two limitations. The first limitation is that the model decrease in performance can be accounted to the distribution shift between the length of the train and the test data, as the baseline model is trained on document-level dataset, in which the premise and hypothesis pairs are much longer. The second possible limitation arise from the fact that appending a random sentence to the premise makes the premise look less natural and create noises for the model. This can also contribute the drastic drop in performance of the models. To conclude, distribution shift in length and noises might contribute to the drastic drop in model performance besides the biases which the model has learned from the training dataset of DocNLI.

Subcase of HANS	**HANS**	**appended HANS**	**Label**	**Subcase**
ln_subject/object_swap	1.00	-	not-entail	Lexical
ln_preposition	1.00	-	not-entail	Lexical
ln_relative_clause	1.00	-	not-entail	Lexical
ln_passive	1.00	-	not-entail	Lexical
ln_conjunction	1.00	-	not-entail	Lexical
sn_NP/S	1.00	-	not-entail	Subsequence
sn_PP_on_subject	1.00	-	not-entail	Subsequence
sn_relative_clause_on_subject	1.00	-	not-entail	Subsequence
se_PP_on_obj	1.00	-	entail	Subsequence
sn_past_participle	0.98	-	not-entail	Subsequence
sn_NP/Z	0.97	-	not-entail	Subsequence
le_passive	0.92	-	entail	Subsequence
cn_adverb	0.89	-	not-entail	Consequence
cn_disjunction	0.84	-	not-entail	Consequence
ce_after_since_clause	0.83	-	entail	Consequence
cn_embedded_under_verb	0.80	-	entail	Consequence
ce_adverb	0.68 (↓ 0.19)	0.41 (↓ 0.46)	entail	Consequence
cn_embedded_under_if	0.63 (↓ 0.24)	0.64 (↓ 0.23)	not-entail	Consequence
le_around_relative_clause	0.48 (↓ 0.39)	0.35 (↓ 0.52)	entail	Lexical
ce_embedded_under_since	0.44 (↓ 0.43)	0.36 (↓ 0.51)	entail	Consequence
le_relative_clause	0.43 (↓ 0.43)	0.37 (↓ 0.50)	entail	Lexical
ce_embedded_under_verb	0.24 (↓ 0.63)	0.38 (↓ 0.49)	entail	Consequence
se_understood_object	0.19 (↓ 0.68)	0.37 (↓ 0.50)	entail	Subsequence
se_relative_clause_on_obj	0.19 (↓ 0.68)	0.35 (↓ 0.52)	entail	Subsequence
ce_conjunction	0.18 (↓ 0.69)	0.36 (↓ 0.51)	entail	Consequence
se_adjective	0.14 (↓ 0.73)	0.41 (↓ 0.46)	entail	Subsequence
le_around_prepositional_phrase	0.10 (↓ 0.77)	0.36 (↓ 0.51)	entail	Lexical
cn_after_if_clause	0.60 (↓ 0.27)	0.61 (↓ 0.26)	not-entail	Consequence
se_conjunction	0.30 (↓ 0.57)	0.37 (↓ 0.50)	entail	Subsequence
le_conjunction	0.00 (↓ 0.87)	0.37 (↓ 0.50)	entail	Lexical

Table 4.4: Accuracy of the baseline model RoBERTa-large trained on DocNLI testset on (1) original *HANS* dataset and (2) DoCNLI testset randomly appended with HANS test instances by subcases ranked by accuracy divided by the upper-half (best resulsts) and the lower-half (worser results).

5 Adversarial dataset construction

The findings of the previous section 4.3 reveals that despite of being trained on a large-scale dataset and at the document-level, which contains much richer contextual relationship. The model trained on DocNLI still performed worse on the sentence-level dataset. It leaves us with the question, whether the diagnostic test results are complete evidence to show that the DocNLI dataset indeed contain biases that leads to a decrease in performance of the model which is trained on it.

For this reason, we carry out further experiments to make sure that our speculation about the bias in the DocNLI is accurate. In particular, we develop a high-quality adversarial dataset to eliminate other factors that can negatively affect the model performance and ensure that only the biases in the data can cause poor performance. As we addressed the limitations of the elementary experiments in the previous section, we constructed an adversarial dataset which contains simple heuristics to reveal the bias in the data such that if the model learned and relies on the superficial heuristics to make prediction, it will fail in our adversarial dataset. Instead of creating new challenge set from scratch, we create it by augmenting existing test instances, which requires less effort to create, and thus the challenge set remains close to the original data distribution. With this approach, we can make sure that if the model decreases in performance, it will not be affected by a grammatical error or the unnaturalness of the appended adversarial sentence within the context of the original test instance.

Section 5.1 discuss our method to construct the adversarial testset and the reason we believe it would mitigate the problem with the two tests from the previous chapter. Section 5.2 describes step for step how the adversarial dataset was created. Section 5.3 provides some key statistics of our newly created dataset. Finally, section 5.4 reports the performance of models trained on DocNLI on our adversarial testset.

5.1 Preliminary

Although NLI models show their weakness in our simple diagnostic test set, we acknowledge the limitation of this approach. Our goal in this chapter is to create a naturalistic adversarial testset to ensure that the model performance drop is due to the biases present in the training dataset and not the fake premise and hypothesis pairs. One way to achieve this is to manually write the premise and hypothesis for the adversarial testset which are similar to the test instances in the DocNLI dataset. However, considering the length and the complexity of document-level NLI dataset, it would be impossible for us to manually create enough test instances and guarantee the quality of the adversarial testset. For this reason, we would choose another approach that would allow us to generate a lot of test instances with high quality and can capture different types of biases.

As shown in the previous section, the performance of the models trained on the DocNLI dataset seems to decrease in the HANS and ANLI testset, one viable solution is reusing this dataset to augment the DocNLI testset into the similar structure of these two adversarial datasets. Subsequently, we consider the feasibility of this approach by looking into the way the HANS and ANLI were created. The controlled evaluation set HANS (McCoy et al., 2019) was created by applying templates based on their list of heuristics including lexical overlap heuristics, subsequence heuristics, and constituent heuristics. On the other hand, ANLI (Nie et al., 2020) was generated via human-in-the-loop training. In essence, writers were asked to write test instances that they think would fool the models with rationales explaining why the model would fail. Another group of annotators are asked to verify the correctness of the test instances. These test instances are used to trained the models. This process is repeated with three different models to general the final version of the dataset. It is clear to see that it would be impossible to follow ANLI approach to create our own version of the adversarial dataset because it requires many annotators, which is out of our capability within this work. By this end, we would opt for augmenting our dataset based on the given templated from the HANS paper by (McCoy et al., 2019) for simplification.

5.2 Dataset contruction

In this section, we describe in detail the method used to generate our adversarial dataset. The input of our method is the original testset of any NLI dataset and a list of templates, which is used to create the HANS dataset McCoy et al. (2019). In general, we select one sentences in the premise and apply the template from HANS to create the augmented hypothesis. Then, we replace the augmented premise and hypothesis sentences with the original premise and hypothesis sentences respectively. Our data augmentation consists of three main steps as follows:

1. Choose only samples that have the entailment label
2. Filter the sentences in the premise that has not been used in the hypothesis
3. Apply the templates to these candidate sentences

In the first filter step, we specifically choose premise-hypothesis pairs that have entailment labels to control the label after the augmentation. The intuition behind this filtering step is that we want to avoid the complex causal change in the augmented sample. Because the DocNLI does not provide any evidence to explain which sentence the evidence that leads to the entailment or non-entailment, or the fine-grained relationship between each of the sentences in the premise and the hypothesis. One potential pitfall of selecting a contradictory sample is that we can by chance choose a sentence in the premise, which leads to the contradiction of the premise and the hypothesis. By augmenting this sentence, we can change it in a way that it might not contradict to hypothesis anymore, thus, involuntarily changing the label. On the other hand, if we only use samples with entailment labels, we can make sure that every sentence in the hypothesis entails the premise. Thus augmenting a sentence in the hypothesis to make it contradict a sentence in the premise would definitely lead to a flip of the label from entailment to contradict.

Next, we filter out the premise sentences that do not have any overlapping with the hypothesis, which means they have not been paraphrased in the hypothesis. The motivation behind this is that we only want to add an augmented sentence from the premise, which does not contribute to the entail/contradict relationship between the premise and the hypothesis to ensure that we do not by chance flip the label. It is worth mentioning that with the size of the data and the size of the premise and hypothesis pair it would infeasible to accomplish this manually to select out white-list premise sentences with 100% accuracy. In this case, we only aim for near-to-perfect white-list premises. To achieve this soft-matching between the sentence in the premise and the hypothesis, we consider the similarity between the two sentence representations. We consider two sentences two be similar if their sentence embedding is close to each other in the embedding space. In particular, we utilize the sentence-transformers[1] (Reimers and Gurevych, 2019) to first compute the sentence embedding and then compare the similarity of each pair sentence in the premise and the hypothesis with cosine similarity. For each of the sentences in the

[1] https://www.sbert.net/docs/usage/semantic$_textual_similarity.html$

hypothesis, we choose a sentence with the highest similarity in the premise as its perfect match. In the end, we have a matching map for each of the sentence in the hypothesis. We later use this map to filter out sentences in the premise that are paraphrased in the hypothesis. The result of this filtering step is a list of non-paraphrased sentences or a white-list sentences of all premises. By performing this filtering step along with the previous filtering step, we end up having a subset of the original DocNLI testset and for each of the instances, we have their white-list premise sentence that we can use for augmenting in the next step.

In the third step, we reuse a list of templates that can be used to generate adversarial samples by McCoy et al. (2019). We will discuss here in detail how we construct each subcase from the HANS templates to augment the white-list premise sentences that are selected from the second step. These templates we used are provided in the HANS paper. In order to use their templates, we need to identify several grammatical components in an ORIGINAL PREMISE and ORIGINAL HYPOTHESIS such as noun phrases, subjects, and main verbs. Particularly, we allocate the noun phrase and the verbs by employing a constituency parser on each of the sentences in the premise and a selected sentence that satisfies that requirement. By using the results from the constituency parser, we apply our special rules to select our sentences that have a subject which is a noun phrase. Particularly, we utilize benepar[2] implementation of the Berkeley Neural Parser to create a constituency a constituency parse tree for each candidates sentence from the previous step.

Another challenge in creating adversarial data from a template is that in some cases we have to generate additional text instead of manipulating the word order or adding some dummy connection words. It would be also very time intensive to manually write this additional text, thus we consider using a generative language model to perform this step for us since there are many language models available that are able to create high-quality text. In particular, we use GPT-2 implementation from Hugging Face[3] to generate the sentence using a seed of the noun phrase. For example, we would like to create a sentence followed this template *[NP1 V1 NP2 who NP3 V2 NP4]* or "The actors called the banker who the tourists saw". First, we use the seed sentence from the seed sentence s_{seed} from premise "The actors called the banker". Then we concatenate this phrase s_{seed} with the word "who" to the GPT-2 model to generate the rest of the sentence.

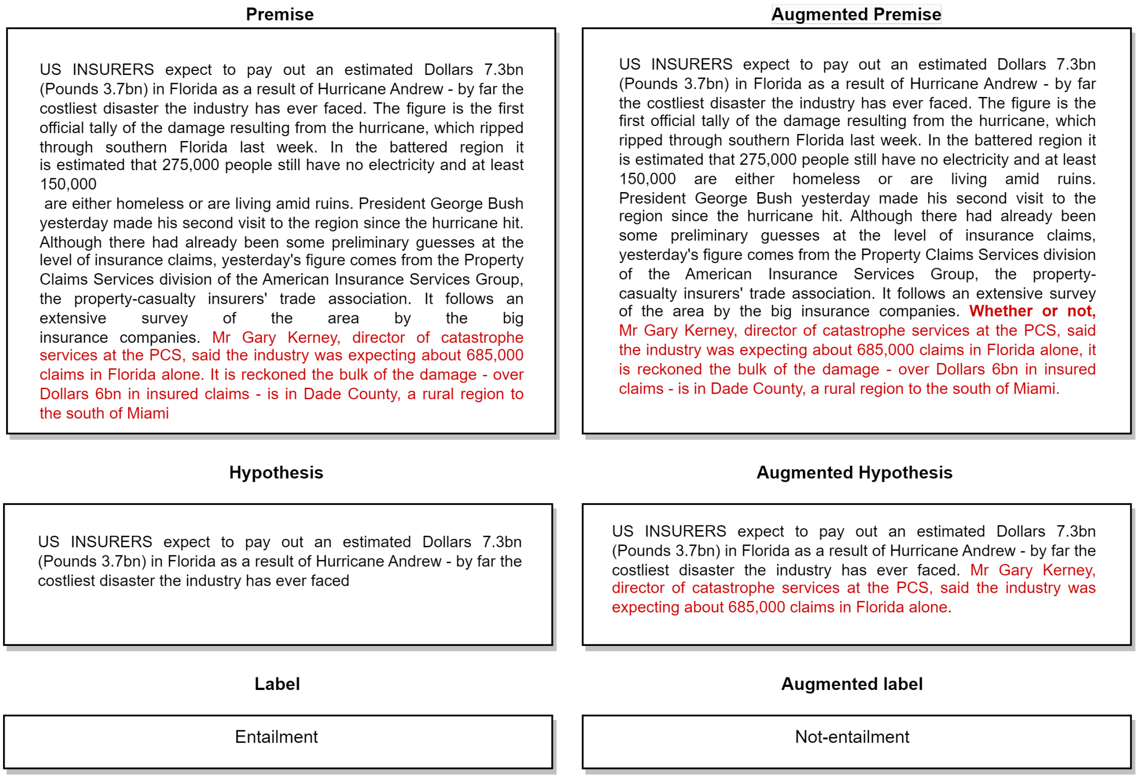
After having the augmented instances of the adversarial testset. We additionally perform post-editing, which is to manually check the generated examples. For each generated

[2] https://pypi.org/project/benepar/

[3] https://huggingface.co/gpt2

instance, we carefully checks for two criteria: the grammatical correctness whether the noun-verb-agreement is correct and added corrections if needed, and the contextual correctness between the generated premise and hypothesis sentence.

The advantage of using the template to generate adversarial examples over modifying natural occurrences in the DocNLI testset is that we ensure the plausibility of the generated sentence. Because we have total control for each of the step in the generation process to avoid breaking the original logic in the premise-hypothesis pairs. Moreover, it also provides us a way to generate many test samples in timely manner, which is beneficial if we want to increase the size of our adversarial dataset.



Premise

US INSURERS expect to pay out an estimated Dollars 7.3bn (Pounds 3.7bn) in Florida as a result of Hurricane Andrew - by far the costliest disaster the industry has ever faced. The figure is the first official tally of the damage resulting from the hurricane, which ripped through southern Florida last week. In the battered region it is estimated that 275,000 people still have no electricity and at least 150,000
are either homeless or are living amid ruins. President George Bush yesterday made his second visit to the region since the hurricane hit. Although there had already been some preliminary guesses at the level of insurance claims, yesterday's figure comes from the Property Claims Services division of the American Insurance Services Group, the property-casualty insurers' trade association. It follows an extensive survey of the area by the big insurance companies. Mr Gary Kerney, director of catastrophe services at the PCS, said the industry was expecting about 685,000 claims in Florida alone. It is reckoned the bulk of the damage - over Dollars 6bn in insured claims - is in Dade County, a rural region to the south of Miami

Augmented Premise

US INSURERS expect to pay out an estimated Dollars 7.3bn (Pounds 3.7bn) in Florida as a result of Hurricane Andrew - by far the costliest disaster the industry has ever faced. The figure is the first official tally of the damage resulting from the hurricane, which ripped through southern Florida last week. In the battered region it is estimated that 275,000 people still have no electricity and at least 150,000 are either homeless or are living amid ruins. President George Bush yesterday made his second visit to the region since the hurricane hit. Although there had already been some preliminary guesses at the level of insurance claims, yesterday's figure comes from the Property Claims Services division of the American Insurance Services Group, the property-casualty insurers' trade association. It follows an extensive survey of the area by the big insurance companies. **Whether or not,** Mr Gary Kerney, director of catastrophe services at the PCS, said the industry was expecting about 685,000 claims in Florida alone, it is reckoned the bulk of the damage - over Dollars 6bn in insured claims - is in Dade County, a rural region to the south of Miami.

Hypothesis

US INSURERS expect to pay out an estimated Dollars 7.3bn (Pounds 3.7bn) in Florida as a result of Hurricane Andrew - by far the costliest disaster the industry has ever faced

Augmented Hypothesis

US INSURERS expect to pay out an estimated Dollars 7.3bn (Pounds 3.7bn) in Florida as a result of Hurricane Andrew - by far the costliest disaster the industry has ever faced. Mr Gary Kerney, director of catastrophe services at the PCS, said the industry was expecting about 685,000 claims in Florida alone.

Label

Entailment

Augmented label

Not-entailment

Figure 5.1: Augmentation example

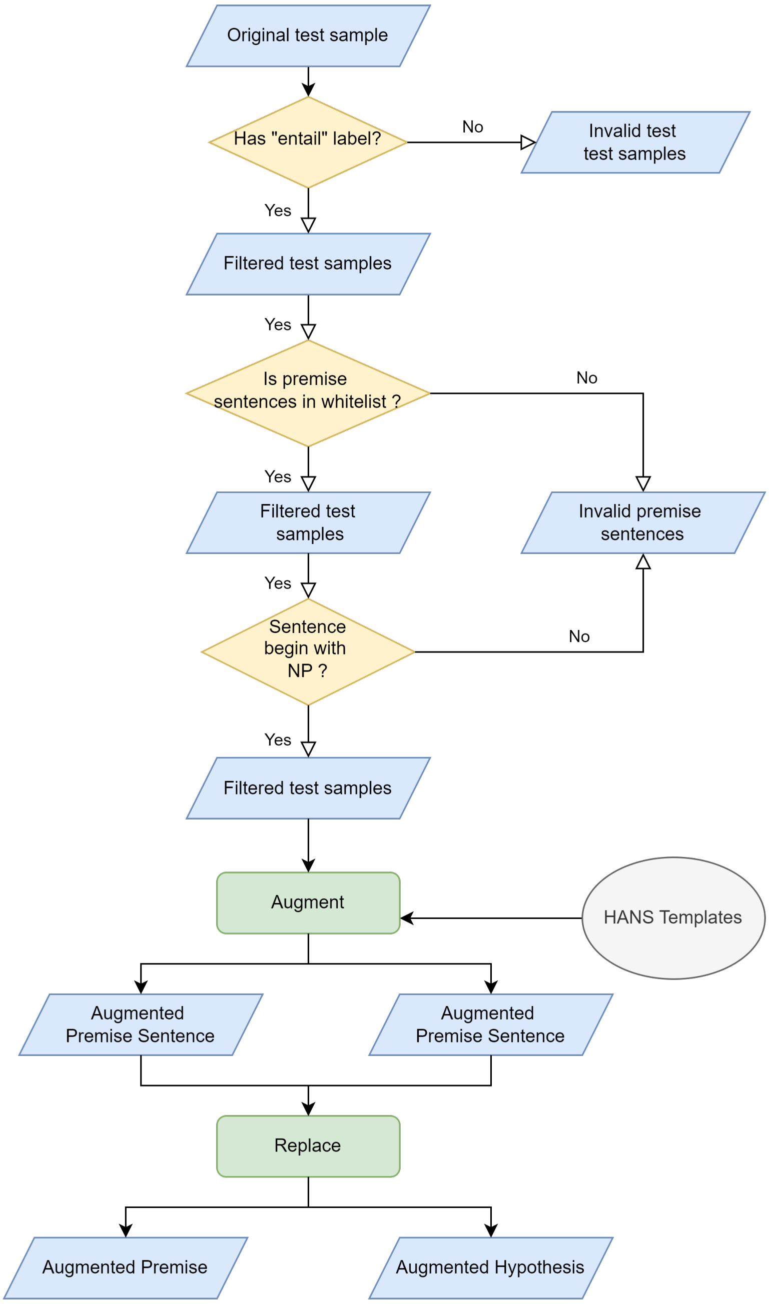


Figure 5.2: Our process to create challenge test set by augmenting the existing DocNLI testset

Number of sample per case	100
Number of cases	11
Number of subsequence cases	2
Number of lexical-overlap cases	3
Number of constituent cases	6
Number of entail cases	200
Number of not-entail cases	900

Table 5.1: Data statistics of our adversarial dataset

5.3 Data statistics

We use the methods described in the previous section and apply them with the templates for data augmentation in HANS paper (McCoy et al., 2019) using all of the data from the test dataset of DocNLI. Table 5.1 shows the statistics of our augmented dataset. For each of the subcases, we generate 100 augmented samples from 100 random samples chosen from the DocNLI testset. In case, we have to discard a sample in post-editing, we will generate other sample to replace it. Due to time constraints, we could not generate and verify the for all of the 23 subcases in the HANS diagnostic dataset. Instead of that, we choose 11 cases where the accuracy of testing on the easy appending test has the biggest drop, as shown in the results in Table 4.4. In total, we have 1100 sentences that are automatically created and manually check. In the check process, the augmented premise and the hypothesis are being checked for grammatical correctness and logical correctness. The main source of the problem is from the noun phrase module, for example “Hurricane Andrew” is recognized wrongly as a person because it contains a person’s name. These types of errors are removed from our dataset to create a high-quality version to make sure that the grammatical error won’t affect the model’s accuracy.

5.4 Results and discussion

In this section, we carry out experiments with our newly generated adversarial test set and report the results. We use the baseline models RoBERTa-large and Longformer and test on our adversarial test set and report the results in Table 5.2. We notice that for 8 out of the 11 subcases the model does not perform as well as in the in-distribution test, which is lower than the accuracy on the original test set. In particular, for the majority of the

subcases	#samples	Types of heuristics	RoBERTa-large	Longfomer-base
baseline			0.87	0.69
ce_adverb	100	constituent	0.04 (↓ .83)	0.07 (↓ .662)
cn_embedded_under_if	100	constituent	0.89 (↑ .02)	0.87 (↑ .18)
le_around_relative_clause	100	lexical-overlap	0.16 (↓ .71)	0.20 (↓ .49)
ce_embedded_under_since	100	constituent	0.10 (↓ .77)	0.08 (↓ .61)
le_relative_clause	100	lexical overlap	0.20 (↓ .67)	0.35 (↓ .34)
ce_embedded_under_verb	100	constituent	0.03 (↓ .84)	0.10 (↓ .59)
ce_conjunction	100	constituent	0.11 (↓ .76)	0.15 (↓ .54)
se_adjectiv	100	subsequence	0.04 (↓ .83)	0.12 (↓ .57)
le_around_prep_phrase	100	lexical-overlap	0.11 (↓ .76)	0.19 (↓ .50)
cn_after_if_clause	100	constituent	0.92 (↑ .05)	0.95 (↑ .26)
se_conjunction	100	subsequence	0.10 (↓ .77)	0.18 (↓ .51)
total	1100			
avg. accuracy			0.25 (↓ .062)	0.30 (↓ .39)

Table 5.2: Accuracy of the models based on subcases of our adversarial test set

subcases, there is a significant drop in the accuracy of the model and the average over all cases, with the greatest decrease of 0.83 in a subcase. However, there is a pecularity with the *not-entail* subcases *cn_embedded_under_if* and *cn_after_if_clause*, because the model perform better than the baseline. As we discuss earlier in Section 4.2.3, this can be explained by that fact the models recognized the *not-entail* better than the *entail* due to the flaw in the construction of the DocNLI dataset.

These results correlate with the results from the previous section with simple data augmentation techniques of simply appending a random challenge sentence at the end of the premise and hypothesis to trick the model. Our result shows that the RoBERTa model and the Longformer model trained on the DocNLI indeed suffer from the syntactic heuristics from the DocNLI dataset. These results would suggest that the poor performance on the adversarial test set is caused by the training data. With our newly created adversarial dataset, we can exclude the other factors that can affect the model performance such as distribution shift in length and noises.

6 Debiasing methods

We conclude from the experiments of the previous chapter that the model does expose weaknessé in the syntactical adversarial test. The primary goal of this chapter is to determine whether debiasing techniques, which are proven to be effective at sentence-level NLI debiasing can help us mitigate biases problem at the document-level. Another goal is to compare the effectiveness of different data-centric and model-centric strategies. In the previous chapter, we could not quantify the biases and identify the biased training instances directly through statistical tests, thus any concrete prior knowledge about the biases in DocNLI is not available. Acknowledging this limitation of our work, we focus only on methods that can remove unknown biases from the datasets. Section 6.1 describes the two data-centric debiasing techniques: *AFlite* and *z-filtering*. Section 6.2 introduces the two model-centric techniques *Product-of-Expert* and *Reweight Instance*. Finally, we evaluate in Section 6.3 the results from these debiasing techniques and provide some insights we made from results.

6.1 Data-centric debiasing methods

In this section, we experiment with two popular methods to filter out biased instances from the DocNLI dataset. Although there are two lines of work that try to alleviate the biases from a dataset, namely *data augmenting* and *data filtering*, we want to focus on the later. The first reason for our choice is that we do not know the exact biased features such that we can add the right data to balance out the effect of this feature. The second reason is that by removing the data samples we can verify our hypothesis about the data bias in an easy manner. If the data is not biased and all data instances are important for the model to learn, training models with less data should reduce the model performance on the adversarial testset and the original DocNLI testset. On the other hand, if the model performance on the adversarial testset does not decrease but improve even though it has been trained on less data, we can conclude that the instances we have removed contain biases that leads to the models heuristics learning. Furthermore, it is not beneficial to follow the data augmentation approach by blindly introducing additional data without knowing the true causes behind the biases. Even if the model performance improves on the adversarial testset we cannot conclude that successfully remove the biases by creating data balancing. Because we do not know the source of improvements: if the models only learn more diverse linguistic phenomena from the additional training instances or if the additional instances cancel out the biased features.

Among the data-centric methods, we choose out *AFlite* and *z-filtering* for further experiments for two reasons: The first reason is that they are simple data filtering methods which are proven effective on a wide range of NLI datasets. The second reason is that these techniques are model-agnostic, hence they are more applicable than other methods. In addition, the advantage of AFlite algorithm is that it requires no manually-written biases. Unlike AFlite, z-filtering requires prior knowledge about the bias in the dataset. Although we cannot quantify the biases in dataset correctly, our diagnostic set gives us an overview of what types of biases are likely to be included in DocNLI dataset, which eventually enables us to define a set of biased features for z-filtering algorithm.

To sum up, our debiasing process with AFlite and z-filtering consists of two steps. First, we emply use AFlite and z-filtering to create the debiased dataset D_{AFlite} and $D_{z-filtering}$. Later, we use these debiased datasets to train the two baseline models (RoBERTa and Longformer). Moreover, we expect to prove our hypothesis in this chapter as follows: If after removing the potentially biased training instances the model accuracy increases, we can prove that the data indeed include biases. We believe that this hypothesis lay a good foundation for all of our experiments in the following sections.

6.1.1 AFlite

Algorithm

Sakaguchi et al. (2019) initially propose to AFlite to remove annotations artifacts from Winograd dataset of the Winograd Schema Challenge to create its debiased version, the WINOGRANDE dataset. Later, Le Bras et al. (2020) performed controlled experiments and proved that AFlite is indeed effective in removing spurious correlations from machine learning datasets problems including NLI (SNLI) and image recognition (ImageNET). This algorithm was also employed to debias other datasets such as ReCoRD (Zhang et al., 2018b), DROP (Dua et al., 2019), HellaSWAG (Zellers et al., 2019), NLI (Bhagavatula et al., 2020), and WinoGrande (Sakaguchi et al., 2019). Following their results, we employed AFlite to remove the spurious artifacts from our dataset.

The goal of AFlite is to filter out biased instances based on the assumption that easy-to-predict instances are more prone to be biased, thus we want to remove instances that have right predictions many times with high confidence. The filtering process is accomplished through an iterative process, in which multiple linear classifiers is trained on different subsets of the data. After that for each instance, we calculate the predictability score as the ratio between the number of times it got predicted correctly over its total number of predictions. This procedure is repeated until there are less than k instances that pass the τ threshold or the number of instances left is fewer than n. Algorithm 1 provides the implementation of the AFlite algorithm.

Algorithm 1 AFlite (Le Bras et al., 2020)

```
Input: Dataset $D = (X, Y)$, pre-computed representation $\phi(X)$, model family M, target dataset size n, number of random m, training set size $t < n$, slice size $k \leq n$, early-stopping threshold $\tau$
Output: reduced dataset $S$
$S = D$;
while $|S| > n$ do
    for all $i \in S$ do
        Initialize multiset of out-of-sample predictions $E(i) = \varnothing$;
        for $iterationj : 1..m$ do
            Randomly partition S into $(T_j, S\backslash T_j) s.t. \mid S\backslash T_j \mid = t$;
            Train a classifier $\mathcal{L} \in \mathcal{M}$ on $\{\phi(x), y) \mid (x, y) \in S\ T_j\}$ ($\mathcal{L}$ is typically a linear classifier);
            for $i = (x, y) \in S$ do
                Compute the predictability score $\tilde{p}(i) = \mid \{\hat{y} \in E(i) s.t. \hat{y} = y\} \mid / \mid E(i) \mid$;
                Select up to $k$ instances $S^{'}$ in $S$ with the highest predictability scores subject to $\tilde{p}(i) \geq \tau$;
                $S = S\backslash S'$;
                if $\mid S' \mid < k$ then
                    break
return S
```

Configurations

To measure the effectiveness of AFlite in removing the biased training instances in the original DocNLI dataset, we trained RoBERTa-large (Zhuang et al., 2021) and Longformer-base (Beltagy et al., 2020), our baseline models on our reduced dataset D_{AFlite} and compare it with the performance of the baseline model on the original DocNLI dataset $D_{baseline}$. For the classifier M, we choose a logistic regression as our linear classifier. We set the hyperparameters $m = 65000$, $n = 20$, $k = 100000$, and $\tau = 0.75$. m is the number of random partitions, n is the training set size, k is the number of instances with the highest score for each slice, and τ is the early stopping threshold. To speed up the computation, we only feed the pre-calculated embeddings to the logistic models for training and inference as the representation of the premise-hypothesis pairs. To obtain the pre-calculated embeddings, we calculate the representation of each training instance by averaging the embedding of individual tokens in the premise and hypothesis. then, we concatenate the embedding of premise and hypothesis together to get an embedding of size $300x2 = 600$. To obtain the token embedding, we first tokenize the premise and hypothesis, then we perform stemming on all tokens. Finally, we get the embedding of the stemmed tokens from fastText model cc.en.300.bin[1]. After running the AFlite on the DocNLI training set, we receive a hard subset D_{AFlite} containing 640039 instances, which equals to one-third of the original DocNLI dataset $D_{baseline}$.

6.1.2 Z-filtering

Algorithm

z-filtering is an adversarial filtering algorithm proposed by Wu et al. (2022) to remove data biases from the SNLI and MNLI dataset. Their algorithm is based on the statistical techniques proposed by Gardner et al. (2021) to measure the spurious correlations between features of the samples and their labels, namely *z-statistics*. Z-statistics measures the deviation of a data point from the mean of the dataset. Algorithm 2 provides an illustration of the z-fitering algorithm. The input is the original dataset $\mathcal{D}'$. At each iteration, our goal is to filter out a set of biased features $\mathcal{B}(l)$ of the partially built debiased subset $\mathcal{Z}$. First, we calculate the z-statistic of each feature for each instance. Based on the z-statistics values, we can decide which features is biased, and thus can remove instances that contain those biased features. These biased instances are added into the biased subset $\mathcal{Z}^{-}(\mathcal{D})$. The un-biased samples are kept in the debiased subset $\mathcal{Z}(\mathcal{D})$. This iteration is repeated over all subset of $\mathcal{D}'$. In particular, the z-statistic of an instance for each feature can be

[1] https://fasttext.cc/docs/en/crawl-vectors.html

calculated as follows:

$$z^*(x,l) = \frac{\hat{p}(l \mid x - p_0)}{\sqrt{p_0(1-p_0)/n}} \tag{6.1}$$

In a unbiased dataset, we expect there is no correlation between each of the features and the class labels. For all feature x in our feature set $\mathcal{X}$, $p(l \mid x)$ should be uniform over the class label l, where $\hat{p}(l \mid x) = \frac{1}{n}\sum_{i=1}^{n} l^j$ is the empirical expectation of $p(l \mid x)$ over n sample containing x and p_0 is the probability of uniform distribution ($p_0 = 0.5$ for NLI with two labels). We consider four different features for our feature set $\mathcal{X}$:

- the percentage of the exact word-overlap between the premise and the sentence
- the boolean value of whether the hypothesis is a sub-sequence of the premise
- the length of the hypothesis
- the ratio between the premise and the hypothesis length

We choose the two first features because they are representative for the biases we found in the adversarial test. Our adversarial testset contains the word-overlap, subsequence, and constituent heuristics. The two last features are added because based on prior works the length of the premise and hypothesis can also be hidden feature revealing the labels to the models.

Configurations

We reimplement the z-filtering algorithm and the feature extraction. Then, we feed the original training set of DocNLI into the z-filtering algorithm over with the number of batch $n = 20$ and the batch size of $k = 45000$. After being filtered, we obtain a debiased subset $\mathcal{Z}(\mathcal{D})$ with 360 000 training instances. We then use this subset in further experiments in this sections.

6.2 Model-centric debiasing methods

In this section, we attempt to mitigate the data bias problem from the model perspective. In particular, we aim to experiment with the models that can generalize well despite many NLI datasets containing specific syntactical heuristics. There have been a lot of

Algorithm 2 Z-filtering (Wu et al., 2022)

```
Input: input dataset $D_0$ [ with optional seed dataset $D_{seed}$ ]
Output: debiased dataset $\mathcal{Z}$ and the rejected samples $\mathcal{Z}^-$
$\mathcal{Z} \leftarrow \varnothing$ (or $\mathcal{Z} \leftarrow D_{seed}$)
$\mathcal{Z}^- \leftarrow \varnothing$
for sample batch $D'_t \subset D'$ do
    compute or update z-statistics $z^*(x, l \mid \mathcal{Z}), x \in \mathcal{X}$ of $\mathcal{Z}$;
    find the biased features $\mathcal{B}_{\mathcal{Z}}(l)$, $l \in$ { entailment, neutral, contradiction };
    for each instance $I = (P, H, l) \in D'_t$ do
        get the features f of the instance $I$;
        if $f \cap \mathcal{B}_{\mathcal{Z}}(l) = \varnothing$ then
            $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{I\}$;
        else
            $\mathcal{Z}^- \leftarrow \mathcal{Z}^- \cup \{I\}$;
```

state-of-the-art debiasing methods developed with remarkable results on the benchmark test sets. However, these methods are so far only dedicated to sentence-level dataset such as MNLI and SNLI. In the scope of this work, we verify their effectiveness in a different domain especially the document-level dataset DocNLI. Among the model-centric debiasing methods, we select two methods that follow a self-debiasing approach that do not require prior knowledge about the biases, namely *reweight instances* (Clark et al. (2019) , Utama et al. (2020a), Utama et al. (2020b)) and *product-of-expert*.

6.2.1 Methodology

Our approach follows the self-debiasing framework suggested by Utama et al. (2020b). It comprises of two steps:

- automatically identifying biased examples using a shallow model f_b
- using this information to train the main model f_d with a debiasing training objectives $\mathcal{L}_d$

In particular, we trained a shallow model f_b on only a subset of dataset. Then we use this biased model to make predictions on the remaining *unseen* examples $\{x^i, y^i\}$ and obtain the probabilities for each class $f_b(x^i) = p_b^i$. Utama et al. (2020b) theorize that the probabilities p_b^i indicates how likely the training instance contain bias. To be concrete, in case of a correct prediction p_b^{i+}, the more confidence the model is about its prediction $p_b^{i+} \rightarrow 1$, the more likely it this instance is biased . On the opposite this, if $p_b^{i+} \rightarrow 0$ the model is less confident about its prediction, this instances is likely to be a hard example that we should focus on. Next, we present the two debiased loss in detail.

6.2.2 Reweight instance

After calculating obtaining additional knowledge of each training instances p_b^i, we use this to train the main model f_d. For reweight instance method, we can use p_b^i in calculating the penalty weight term $1 - p_b^i$. In this fashion, we emphasize more on hard training instances, the lower the confidence prediction from our pre-calculation the more we want to penalize the models.

$$\mathcal{L}(\theta_d) = -(1 - p_b^{i+})y^{(i)} \cdot \log p_d \tag{6.2}$$

Product-of-Expert

With Product-of-Expert method, we freeze the weight of the trained shallow model and combine it with the main model by using their softmax output in the custom loss function, which can defined as follows:

$$\mathcal{L}(\theta_d) = -y^i \cdot \log softmax(\log p_d + \log p_b) \tag{6.3}$$

6.2.3 Configurations

We implement the self-debiasing framework suggested by Utama et al. (2020b). The configuration of the baseline model (RoBERTa and Longformer) remains unchanged and only the original cross entropy loss is replaced with the two debiasing loss functions we have defined above. We train our self-debiasing framework in two separate train steps. First, we train a shallow model f_b which is a RoBERTa-base model with the hidden layer size of 768, which is much smaller than the RoBERTa-large and the Longformer-base model with hidden layer size of 1024. Then, we train it on only 1/3 of the original training data of DocNLI, this subset is selected randomly with data shuffling to ensure that the subset is representative for all types of premise-hypothesis pairs in DocNLI training set. This model f_b serves as the weak leaner in our configurations, the main model remains to be the original RoBERTa-large and Longformer-base model. Finally, we train the main models with the debiasing loss for 5 epochs. We reuse the implementation of the debiasing loss functions from (Utama et al., 2020b)[2]. During inference time, we only use the main model RoBERTa-large to predict our custom challenge testset.

6.3 Results

In this section, we present the evaluation of the fine-tuned RoBERTa-large and the Longformer-base model on different debiasing methods. Table 6.2 presents the evaluation on the RoBERTa-large based model, here, it can be observed that the AFlite is the best debiasing methods and Reweight Instance the best worst one. In particular, AFlite outperforms other methods in terms of the average accuracy as well as by subcases. Its average accuracy is also higher than the baseline RoBERTA-large based model. Considering the subcases, it reaches the highest accuracy 9 our of 11 subcases. In contrast, Reweight Instance has the lowest improvements for 10 out of 11 subcases and its average accuracy is 0.24, which ends worst among all methods and is only slightly higher over the baseline. Taking the average accuracy over all subcases into consideration, we conclude that all the data-centric methods (AFlite and z-filtering) outperform the model-centric methods (Product-of-Expert and Reweight Instance). Because the improvements made by the data-centric methods are substantially big enough, while the improvements of model-centric are only marginally higher than the baseline.

Next, we consider the evaluation of the Longformer-based models being trained on the debiased dataset with AFlite and z-filtering techniques in Table 6.3. Overall, both data-centric methods enhance the original dataset such that there are visible improvements of the models trained on the debiased dataset over the baseline. Comparing between two methods, AFlite performs better than z-filtering with a higher average accuracy as well as by subcases. Another key observation is that AFlite is the most effective method for both of the Longformer-based and the RoBERTa-based model, followed up by the z-filtering method.

Our results with the data filtering techniques align with the results by Mishra and Sachdeva (2020). In their work, they reduce the size of the SNLI dataset to 2% of the original SNLI and show that RoBERTA-based model trained on this pruned training set achieves near-equal performance on the SNLI devset and competitive zero-shot generalization on three OOD datasets. Thus, it proves that reducing the size of the dataset does not necessarily

[2] https://github.com/UKPLab/emnlp2020-debiasing-unknown

	D_{AFlite}	$D_{z-filtering}$	D_{DocNLI}
Longformer	0.69	0.67	0.69
RoBERTa-large	0.88	0.86	0.87

Table 6.1: Comparisons between the accuracy of the baseline models and models that are trained on the debiased dataset on the original DocNLI testset

reduce the performance of the models, since big datasets might contain noisy and biased samples that distract the model. By removing these biased samples, we can increase the performance of the models without having to waste a lot of computation into training models on large-scale dataset.

So far, we have evaluate the debias methods on the adversarial testset, however, there is another important aspect needed to be considered. Concretely, we would like to know the effect of the debiasing method to the model performance on in-of-distribution testset. We conducted evaluation on the models trained on the two debiased dataset AFlite and z-filtering methods on the original testset of DocNLI. We can see in Table 6.1 that the data-centric debiasing methods do not hurt the performance of the models on the in-distribution testset. We see that AFlite and z-filtering are helpful to remove the unknown biases from the dataset, which eventually leads to better performance in the out-of-distribution testset (our adversarial testset). We can conclude that the model is not forced to learn hard examples from the debiased testset but also learn sufficient examples to solve the original testset.

Subcase	PoE	Reweight	AFlite	Z-filtering	DocNLI
ce_adverb	0.22 (↑ 0.18)	0.10 (↑ 0.06)	**0.96** (↑0.92)	0.51 (↑ 0.21)	0.04
cn_embedded_under_if	0.58 (↓ 0.31)	**0.91** (↑ 0.02)	0.16 (↓0.73)	0.60 (↓ 0.61)	0.89
le_around_relative_clause	0.29 (↑ 0.13)	0.11 (↓ 0.05)	**0.89** (↑0.73)	0.64 (↑ 0.09)	0.16
ce_embedded_under_since	0.29 (↑ 0.19)	0.09 (↓0.01)	**0.79** (↑0.60)	0.61 (↑ 0.23)	0.10
le_relative_clause	0.23 (↑ 0.03)	0.11 (↓0.09)	**0.55** (↑0.35)	0.15 (↑ 0.05)	0.20
ce_embedded_under_verb	0.30 (↑ 0.17)	0.06 (↑ 0.03)	0.84 (↑0.81)	**0.88** (↑ 0.17)	0.03
ce_conjunction	0.32 (↑ 0.21)	0.11	**0.76** (↑0.65)	0.58 (↑ 0.16)	0.11
se_adjective	0.20 (↑ 0.16)	0.07 (↑ 0.03)	**0.96** (↑0.92)	0.11 (↑ 0.07)	0.04
le_around_prep_phrase	0.24 (↑ 0.13)	0.09 (↓0.02)	**0.70** (↑0.59)	0.64 (↑ 0.25)	0.11
cn_after_if_clause	0.75 (↓ 0.17)	**0.93** (↑ 0.01)	0.28 (↓0.64)	0.30 (↓ 0.47)	0.92
se_conjunction	0.32 (↑ 0.22)	0.07 (↓0.03)	**0.82** (↑0.70)	0.23 (↑ 0.13)	0.10
avg	0.34	0.24	**0.70**	0.48	0.25

Table 6.2: Result of testing RoBERTa-based model after being treated with different debias methods on our adverarial testset. **bold** indicates the maximum values among all methods, underline shows minimum values values among all methods.

Subcase	**PoE**	**Reweight**	**AFlite**	**Z-filtering**	**DocNLI**
ce_adverb	-	-	**1.0** (↑0.92)	0.51 (↑ 0.21)	0.07
cn_embedded_under_if	-	-	0.0 (↓ 0.61)	**0.27**	0.87
le_around_relative_clause	-	-	**0.95** (↑0.73)	0.68 (↑ 0.09)	0.28
ce_embedded_under_since	-	-	**1.0** (↑ 0.23)	0.73	0.08
le_relative_clause	-	-	**1.0** (↑0.35)	0.68 (↑ 0.05)	0.35
ce_embedded_under_verb	-	-	**1.0** (↑0.81)	0.95 (↑ 0.17)	0.10
ce_conjunction	-	-	**0.94** (↑0.65)	0.57 (↑ 0.16)	0.15
se_adjective	-	-	**1.0** (↑0.92)	0.71 (↑ 0.07)	0.12
le_around_prep_phrase	-	-	**0.98** (↑0.59)	0.74 (↑ 0.25)	0.19
cn_after_if_clause	-	-	0.05 (↓0.64)	0.32 (↓ 0.47)	0.95
se_conjunction	-	-	**0.82** (↑0.70)	0.80 (↑ 0.13)	0.18
avg	-	-	0.79	0.63	0.30

Table 6.3: Result of testing Longformer-based models after being treated with different debias methods on our adverarial testset. **bold** indicates the maximum values among all methods, underline shows minimum values values among all methods.

7 Conclusions

In this work, we have demonstrated that the document-level DocNLI dataset suffers from data bias problem. This work to the best of our knowledge, is the first work to extensively investigate the bias problem in a document-level NLI dataset. We conduct different experiments to show the models trained on DocNLI training set are not robust towards adversarial test instances. We find evidences showing that training data of DocNLI contains syntactical heuristics, which eventually lead to the model decrease in performance in adversarial test cases. However, contradicted to our early speculation, we have shown that DocNLI dataset does not contain *hypothesis-only* bias between certain words in the hypothesis and labels.

To improve the accuracy of our bias identification process, we create our custom adversarial dataset to diagnose whether the models suffers from the heuristics that it was trained on. This custom dataset is constructed from the original testset of DocNLI and the templates for syntactic heuristics by McCoy et al. (2019). Our adversarial testset has not only supported us with the bias diagnostic in DocNLI but also with the benchmarking of different debiasing methods.

Considering the evidence of biases in DocNLI dataset, we experiment with different debiasing methods to improve the quality of the DocNLI dataset. In particular, we aim to employ methods for sentence-level NLI dataset to mitigate the biases in the document-level dataset. Our results shows that sentence-level methods can also be transferred to the sentence-level debiasing task. In our evaluation, the models trained on the debiased dataset obtained through data-centric and the model-centric methods are more robust towards our custom adversarial dataset than the baseline model. Furthermore, the data-centric methods are more effective in removing unknown biases from the dataset. The best debias method AFlite achieves 0.70 accuracy on average on the the advesarial dataset, which equals to 0.45 of relative improvement over the baseline. In addition, our results also show that training the models on the debiased methods does not decrease the value of the in-of-distribution test.

7.1 Future Work

In this work, we examine intensively how models trained on DocNLI suffers from syntactical heuristics from the biased dataset. However, a number of limitations still need to be observed regarding our work. First, we leave it for future work to explore the other types of biases in DocNLI dataset. One possible approach is to extend our adversarial testset with more examples covering further types of biases. Further, we can expand our advesarial dataset to contain all subcases from the HANS datatset.

From the experiments we conducted, it is clear to observe that data-centric methods can effectively remove biases in the dataset. However, it is worth noticing that both methods AFlite and z-filtering require hyperparameters configurations. Due to time constraint and computational limit, we could not search for an optimal configurations. For this reason, we hope that future work can leverage our proposals, for example, by apply efficient search method to determine a set of better hyperparmeters than ours. In addition, as we see both of the data-centric and model-centric methods have brought substantial improvements, it would be beneficial to combine both approaches into a hybrid method. By this way, we can utilize the strengths of both approaches and might be able to achieve better debiasing techniques.

Furthermore, the failures of the models and the success of adversarial filtering is only an initial exploration. Thus, it opens more questions about what kinds of biased features there are in DocNLI. As the matter of fact, it would be very interesting to develop more specific techniques to investigate the filtered-out training instances. Through these investigations we can potentially classify the biased instances into different categories. We can also determine which are the biased features in the premise-hypothesis pairs i.e the ratio between the premise and hypothesis length. We can use these insights about the biases to improve the construction of large-scale document-level NLI datasets in the future.

Bibliography

Belinkov, Y., Poliak, A., Shieber, S., Van Durme, B., and Rush, A. (2019). On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262, Minneapolis, Minnesota. Association for Computational Linguistics.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv:2004.05150*.

Bhagavatula, C., Bras, R. L., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., tau Yih, W., and Choi, Y. (2020). Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Clark, C., Yatskar, M., and Zettlemoyer, L. (2019). Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Clark, C., Yatskar, M., and Zettlemoyer, L. (2020). Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Curation (2020). Curation corpus base.

Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognising textual entailment challenge. In *MLCW*.

Demszky, D., Guu, K., and Liang, P. (2018). Transforming question answering datasets into natural language inference datasets.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. (2019). DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Gardner, M., Merrill, W., Dodge, J., Peters, M., Ross, A., Singh, S., and Smith, N. A. (2021). Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, page 517–520, USA. IEEE Computer Society.

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

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

Joshi, N., Pan, X., and He, H. (2022). Are all spurious features in natural language alike? an analysis through a causal lens.

Karimi Mahabadi, R., Belinkov, Y., and Henderson, J. (2020). End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Kaushik, D., Hovy, E., and Lipton, Z. (2020). Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Koreeda, Y. and Manning, C. (2021). ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Le Bras, R., Swayamdipta, S., Bhagavatula, C., Zellers, R., Peters, M. E., Sabharwal, A., and Choi, Y. (2020). Adversarial filters of dataset biases. In *ICML*.

Liu, H., Cui, L., Liu, J., and Zhang, Y. (2021). Natural language inference in context - investigating contextual reasoning over long texts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13388–13396.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Meissner, J. M., Sugawara, S., and Aizawa, A. (2022). Debiasing masks: A new framework for shortcut mitigation in nlu. abs/2210.16079.

Min, J., McCoy, R. T., Das, D., Pitler, E., and Linzen, T. (2020). Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Mishra, A., Patel, D., Vijayakumar, A., Li, X. L., Kapanipathi, P., and Talamadupula, K. (2021). Looking beyond sentence-level natural language inference for question answering and text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1322–1336, Online. Association for Computational Linguistics.

Mishra, S. and Sachdeva, B. S. (2020). Do we need to create big datasets to learn a task? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 169–173, Online. Association for Computational Linguistics.

Moosavi, N. S., de Boer, M., Utama, P. A., and Gurevych, I. (2020). Improving robustness by augmenting training sentences with predicate-argument structures.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. (2018). Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rudinger, R., May, C., and Van Durme, B. (2017). Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, Valencia, Spain. Association for Computational Linguistics.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. (2019). Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Sanh, V., Wolf, T., Belinkov, Y., and Rush, A. M. (2021). Learning from others' mistakes: Avoiding dataset biases without modeling them. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Schuster, T., Shah, D., Yeo, Y. J. S., Roberto Filizzola Ortiz, D., Santus, E., and Barzilay, R. (2019). Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Stacey, J., Minervini, P., Dubossarsky, H., Riedel, S., and Rocktäschel, T. (2020). Avoiding the Hypothesis-Only Bias in Natural Language Inference via Ensemble Adversarial Training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8281–8291, Online. Association for Computational Linguistics.

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. (2020). Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of EMNLP*.

Trivedi, H., Kwon, H., Khot, T., Sabharwal, A., and Balasubramanian, N. (2019). Repurposing entailment for multi-hop question answering tasks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2948–2958, Minneapolis, Minnesota. Association for Computational Linguistics.

Utama, P., Moosavi, N. S., Sanh, V., and Gurevych, I. (2021). Avoiding inference heuristics in few-shot prompt-based finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020a). Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020b). Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Welleck, S., Weston, J., Szlam, A., and Cho, K. (2019). Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wu, Y., Gardner, M., Stenetorp, P., and Dasigi, P. (2022). Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland. Association for Computational Linguistics.

Yaghoobzadeh, Y., Mehri, S., Tachet des Combes, R., Hazen, T. J., and Sordoni, A. (2021). Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics.

Yin, W., Radev, D., and Xiong, C. (2021). DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. (2019). HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Zhang, C. and Chai, J. (2010). Towards conversation entailment: An empirical investigation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 756–766, Cambridge, MA. Association for Computational Linguistics.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018a). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zhang, S., Liu, X., Liu, J., Gao, J., Duh, K., and Durme, B. V. (2018b). Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

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

Zhuang, L., Wayne, L., Ya, S., and Jun, Z. (2021). A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.