석 사 학 위 논 문
Master's Thesis

Annotation Artifact를 감소시키는 자연어처리 데이터셋의 크라우드소싱 기법

Reducing Annotation Artifacts in Crowdsourcing Datasets for Natural Language Processing

2021

한 동 훈 (韓 東 勳 Han, Donghoon)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

Annotation Artifact를 감소시키는 자연어처리 데이터셋의 크라우드소싱 기법

2021

한 동 훈

한 국 과 학 기 술 원

전산학부

Annotation Artifact를 감소시키는 자연어처리 데이터셋의 크라우드소싱 기법

한 동 훈

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2020년 12월 21일

심사위원장 오 혜 연 (인)

심 사 위 원 김 주 호 (인)

심 사 위 원 차 미 영 (인)

Reducing Annotation Artifacts in Crowdsourcing Datasets for Natural Language Processing

Donghoon Han

Major Advisor: Alice Haeyun Oh
Co-Advisor: Juho Kim

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Daejeon, Korea
December 21, 2020

Approved by

Alice Haeyun Oh
Associate Professor of School of Computing

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MCS 한동훈. Annotation Artifact를 감소시키는 자연어처리 데이터셋의 크라우드소싱 기법. 전산학부 . 2021년. 22+iv 쪽. 지도교수: 오혜연, 김주호. (영문 논문)
Donghoon Han. Reducing Annotation Artifacts in Crowdsourcing Datasets for Natural Language Processing. School of Computing . 2021. 22+iv pages. Advisor: Alice Haeyun Oh, Juho Kim. (Text in English)

초 록

비교적 적은 비용과 큰 확장성을 지닌 크라우드소싱 기법을 활용하여 자연어 처리 데이터셋을 형성하는 사례가 늘고 있다. 그러나 크라우드소싱 기법을 활용하여 데이터셋을 형성할 경우 annotation artifact라는 문제가 발생할 수 있다. Annotation artifact는 크라우드소싱 작업자의 태스크와 무관한 글쓰기 전략으로써, annotation artifact를 지닌 데이터셋으로 학습한 모델은 해당 전략을 학습할 수 있고, 따라서 태스크에 대한 학습이 저해될 수 있다. 이를 해결하기 위한 연구가 꾸준히 진행되고 있으나 크라우드소싱을 통한 데이터셋 형성 과정을 개선하는 연구는 미흡하다. 본 학위논문에서는 annotation artifact를 일으킬 것으로 예측되는 단어의 사용을 크라우드소싱 과정에서 통제함으로써, 간단하지만 효과적인 데이터셋 수집 개선 방안을 제안한다. 아마존 Mechanical Turk 플랫폼에서 자연어 추론 데이터셋을 수집할 때 사용 단어에 제한을 가한 결과, 기존 방식에 비해 단위 태스크 당 19.7초가 더 소요된 반면, 정확도 차이를 바탕으로 측정한 annotation artifact는 9.2% 감소함을 확인했다.

핵 심 낱 말 데이터셋, 편향, 크라우드소싱

Abstract

Many NLP datasets are generated with crowdsourcing because it is a relatively low-cost and scalable solution. One important issue in datasets built with crowdsourcing is annotation artifacts. That is, a model trained with such a dataset learns annotators' writing strategies that are irrelevant to the task itself. While this problem has already been identified and studied, there is limited research approaching it from the perspective of crowdsourcing workflow design. We suggest a simple but powerful adjustment to the dataset collection procedure: instruct workers not to use a word that is highly indicative of annotation artifacts. In the case study of natural language inference dataset construction, the results from two rounds of studies on Amazon Mechanical Turk reveal that applying a word-level constraint reduces the annotation artifacts from the generated dataset by 9.2% in terms of accuracy–gap score at the time cost of 19.7 second increase per unit task.

Keywords datasets, annotation artifacts, crowdsourcing

Contents

List of Tables

List of Figures

Chapter 1. Introduction

In natural language processing (NLP), datasets are often generated to train a model that understands language for diverse tasks such as question answering [1] and identifying logical relationships [2, 3]. One of the widely used approaches when constructing such datasets is crowdsourcing, thanks to its low cost and scalability. In one type of crowdsourcing, workers are asked to write a textual statement to build the dataset when it is difficult to collect a text corpus that fits the purpose of the task [2, 4]. While crowdsourcing has gained popularity in dataset construction, a number of studies have reported that such human-elicited datasets have *annotation artifacts*, a type of dataset bias in which workers' strategies to generate data instances provide a task-irrelevant shortcut to correct prediction [5, 6, 7, 8].

For example, there is a NLP benchmark task named natural language inference (NLI), the goal of which is to correctly classify a pair of statements (so-called premise and hypothesis) according to their logical relationship: *entailment*, *neutral*, or *contradiction*. SNLI, the first large-scale dataset of NLI, was generated with crowdsourcing [2]. Due to the high cost required to collect a number of sentence pairs with clear logical relationships, crowd workers were prompted to write a statement (hypothesis) satisfying a logical relationship given a premise. Table 1.1 shows example sentence pairs in SNLI. One reported case of annotation artifacts in SNLI is the predominant frequency of negation words like *not* in the *contradiction* class, compared to the other two classes [6, 7]. The skewed distribution of a word over classes might give a clue about the correct answer, which likely causes a model to take the shortcut instead of learning logical relationships from the task. In fact, it has been discovered that the high accuracy of a few neural models is attributed to these annotation artifacts [6, 7, 8].

A number of approaches have been proposed to resolve annotation artifacts from the existing NLP datasets. Mostly, they suggest either altering the way that a model is trained [9, 10, 11, 12, 13] or augmenting dataset with adversarial instances [14, 15, 16, 17]. However, research to date has tended to focus on post-hoc solutions. Without fixing the data generation scheme, this problem will keep occurring when constructing a dataset in a similar manner. Thus, the purpose of this research is to improve crowdsourcing workflow design so that a worker cannot use their own strategies that yield artifacts. While previous studies on annotation artifacts [6, 7] mainly explain the issue in terms of word distributions over classes, the impact of word patterns to the generation of annotation artifacts has not been explicitly investigated—to the best of our knowledge. As a result, we attempt to examine the impact of word patterns in generation of annotation artifacts in this paper and leverage the word patterns to mitigate the issue.

We conducted a controlled study on Amazon's Mechanical Turk (MTurk) [1] to examine the impact of word patterns in generation of annotation artifacts. In the study, we recruited 15 unique workers for the NLI data collection task in each of the two different conditions. In condition *Baseline*, we collected the data in almost the same way that SNLI is collected, while in condition *SW* (single-word), workers are instructed to include a given word when writing. The constraint word is chosen based on the data of condition *Baseline*, the semantic meaning of which is regarded as the least associated with the class. Interestingly, a model trained on the data of condition *SW* exhibits a significantly reduced degree of annotation artifacts compared to that of condition *Baseline* from 18.91% to 9.71% in terms of accuracy–gap score, the metric that we devised to measure the degree of annotation artifact. This result implies

[1] https://www.mturk.com/

Premise	A greyhound with a muzzle runs on a racetrack.
Entailment	The dog is running.
Neutral	The greyhound is racing for the rabbit.
Contradiction	The dog is walking around the house.

Table 1.1: The example instances from SNLI dataset [2]. The goal of the task is to correctly classify the logical relationship given a pair of statements.

that certain words are inherently correlated with a specific class, thus the class-specific word patterns can possibly cause annotation artifacts.

Despite the significant reduction of annotation artifacts by introducing a single-word constraint, another issue arises; condition *SW* takes about twice more time than *Baseline* on average. To understand the relationship between task design, data quality, and task time, we collect an additional set of data from 15 workers on MTurk, giving a choice between five words to include in text generation (instead of one) as a constraint. Considering the task designs from previous studies with different degree of freedom in writing task [2, 15], we discover that there exists a trade-off between task time and the degree of annotation artifact. This result indicates that with stricter constriants given to workers, annotators leverage strategies so that annotation artifacts can deteriorate.

Based on the results of our analyses, we propose design considerations for data generation with constraints, which dataset designers can apply when crowdsourcing text datasets with writing.

This research has the following core contributions:

- We provide evidence that the word patterns are attributed to the presence of annotation aritfacts from the dataset collected by crowdsourcing with writing.
- We show that by simply adjusting the task design in that a worker must inclue a constraint word in their writings, annotation artifacts can be significantly reduced from the dataset. However, as a trade-off, the task time increases along with stricter constraints given to the workers.
- We publish the experiment data [2] of 2.7k instances with validation results for replication and further investigation of workers' behaviors in different workflow designs.

[2] https://doi.org/10.6084/m9.figshare.12962480.v3

Chapter 2. Background and Related Work

2.1 Annotation artifacts

An annotation artifact refers to a type of dataset bias generated from the annotators' strategies that give clues to correct prediction while being irrelevant to the task itself. For example, the crowdsourced benchmark datasets of NLI are found to have annotation artifacts. In fact, a few model significantly outperform random classifier on hypothesis classification while both a premise and hypothesis should be provided to correctly infer the logical relationship [6, 7, 8]. There is another benchmark task named story cloze test (SCT), the goal of which is to correctly predict whether the ending is right or wrong given a context. In the crowdsourced datasets of SCT, a neural model succeeds in making a discrimination of endings even when the context is not provided [5, 18]. For the task of visual question answering (VQA), the purpose of which is to predict the correct answer to a question asking about a given image [19], one study investigated the behavior of VQA models and reported that models exhibit a behavior of making a prediction without fully listening to a question or paying attention to an image [20].

There have been a few successful trials at reducing annotation artifacts. First, model-wise approaches have been proposed to train a robust model against the annotation artifact. For example, an ensemble-based method called Product of Experts (PoE) was suggested to debias a model by improving the training loss with a loss of another model leveraging artifacts [9, 11, 13, 12, 10, 21]. Also, previous studies have argued that adversarially augmenting an existing dataset can reduce the annotation artifact [14, 15, 17]. However, these approaches take either huge computational costs or human annotation costs. A phenomenon called bias amplification raises another concern. The bias amplification indicates that the degree of bias is amplified through forward propagation, thus exhibits a larger degree of bias in prediction. Regarding the matter, we argue that reducing annotation artifacts by improving the crowdsourcing workflow would be more efficient compared to those types of approaches.

A similar idea has been proposed in a paper contemporaneous to ours. Bowman et al. showed that instead of writing from scratch, giving annotators pre-built templates for premises and letting them edit reduces the degree of annotation artifacts [22]. Another work constrained annotators not to use a word which improves the diversity of dataset and the robustness of model [23]. They are both similar to our work in that they adjust the data collection protocol to improve the dataset quality. However, this is the first work which applies a word constraint that annotators must use in their writings, in order to reduce the annotation artifact by balancing the word distribution over classes.

2.2 Satisficing in crowdsourcing

While crowdsourcing is extensively used in a variety of applications, a concern regarding the quality of workers' responses often arises because of strategies leveraged by workers. This kind of worker's behavior of taking a shortcut to complete the task with minimal effort is referred to as "satisficing" [24], with which the reliability and quality of worker's responses can be damaged [25, 26, 27]. In the context of data generation, for example, the post-hoc analysis of SNLI revealed workers' strategies, one of which is that modifiers are often added to a premise to make a hypothesis of *neutral* class [6]. Also, when constructing a crowd-generated text dataset, a worker's strategy was so evident that models could identify the worker

given a statement [28]. These studies imply that, when crowdsourcing textual statements, some of the workers satisfice by continuously applying a word-level strategy, which can cause annotation artifact. Also, these observations suggest that a dataset designer carefully take the satisficing behaivor of workers, as the quality of the generated dataset would be greatly threatened otherwise.

In fact, one study introduced a method called Kapcha to prevent respondents' satisficing behavior in online survey, by allowing users to submit a response after a certain waiting period and adding a fade-in effect on a question to attract visual attention [26]. In a similar vein, this research introduces a word-level constraint to prevent the suspicious satisficing behavior: repetition of word-level pattern.

Chapter 3. Study 1

In this section, we describe our empirical study using MTurk to reveal the effect of word patterns on annotation artifacts and present the results of the study. We chose NLI as the domain of our empirical study. Our method differs substantially from previous research using model-based approaches to mitigating annotation artifacts. But the end goal is the same, so we compare our method with the model-based approaches.

3.1 Conditions

3.1.1 Baseline

In the condition *Baseline*, we collect the data in the same way as the SNLI dataset with minor changes. In the SNLI dataset, words like *animals* and *outdoors* used in the example sentence of *entailment* class are overly used in the annotator-generated hypothesis sentences of the same class [6], which raises a concern that users unnecessarily depend on the lexicon of the example sentences, so we changed the example hypothesis sentences to be syntactically similar with minimal addition of new words (Table 3.1).

3.1.2 Single-word constraint (SW)

We hypothesize that the class-specific word patterns are a major cause of annotation artifacts, and that this pattern is present even in a small NLI dataset. In this condition, we start with the small dataset collected in the condition *Baseline*, and we instruct the user to include a specific word, the *constraint word* when writing the hypothesis sentences. Similar to previous work [6, 7], we use pointwise mutual information (PMI) to select the constraint word. PMI given a word w and a class c is defined as follows:

$$PMI(w, c) = log \frac{p(w, c)}{p(w)p(c)}$$

This metric measures the degree of association between a word w and class c; the lower the PMI is, the less frequent the word w is used in the class c than other classes. For example, when the usage of a word is balanced over all classes, the PMI value of the word for each class will be equal to 1/3. During the data collection, a word which is used more than 10 times over all classes and of the lowest PMI value among the words in a class was selected in real time upon request. Thus, the selection of a constraint word depends on the seed data and responses collected prior to the request. When calculating the PMI value, we applied 5-smoothing to avoid zero division error.

	SNLI	Ours
Premise	Two dogs are running through a field.	John is taking a nap on the sofa.
Entailment	There are animals outdoors.	John is sleeping.
Neutral	Some puppies are running to catch a stick.	John is snoring.
Contradiction	The pets are sitting on a couch.	John is playing soccer with Bob.

Table 3.1: The example sentences provided to workers when collecting SNLI and our data.

Premises
A blond woman is standing outside a Modell's store with a large tote bag.
A cop, and two females pose for a picture next to an officers vehicle.
A greyhound with a muzzle runs on a racetrack. A man crossing the street in the rain.
A person staring at a wall that has a bike against it.
A police officer riding a motorcycle.
A woman stands outside of a church door alone.
The mom is getting ready to give her baby a bath.
Two football players are tackling an opposing football player with a referee nearby.
Two ladies all dressed up and partying on the street.
Two men and a woman sit near the front of a bus with religious artifacts around.
Two men dressed up share a toast.
Two men push three wheeled chairs up an inclining road.
Two women are in a kitchen preparing vegetables in a wok.
Woman sunbathing out on sandy beach under umbrella.

Table 3.2: The list of 15 premises that were used in the experiments. These sentences were randomly sampled from SNLI dataset [2]

3.2 Data collection

Based on the description about the data collection procedure of SNLI [3], we reconstructed the data collection interface (Figure 3.1). For each condition, 15 unique participants on MTurk were recruited in a separate batch, with the following qualifications: (1) residents of the U.S., (2) at least 500 HITs approved, and (3) HIT approval rate greater than 98%. The participants of the condition *Baseline* and *SW* were paid \$5 (\$12.57/hr) and \$6 (\$6.76/hr), respectively. A participant was not allowed to join a task of multiple conditions.

Before entering the annotation interface, a participant was provided with a brief introduction about the task and the interface. On the annotation interface, a participant was asked to write a hypothesis statement of each class given a premise. We used 15 premises for the experiment, which were randomly sampled from premises of SNLI dataset prior to the experiment (Table 3.2). All participants were provided with an identical set of premises. Users were not allowed to modify the responses already submitted. Along with the data collection, we manually inspected workers' responses and rejected the worker when more than half of the responses are found to be irrelevant to the task as a way of quality control. For each condition, we kept recruiting workers with this filtering process until the number of workers whose responses are valid reached 15. Following the task, participants were asked to respond to a questionnaire designed to understand the task load and get feedback.

3.3 Data validation

After completing the data collection in each condition, we validated the data on MTurk to filter out invalid instances. Similar to the validation procedure in SNLI [2] and MNLI [3], for each pair of statements we received four labels from annotators. A label represents the relationship between the pair of statements with one of the four categories, including the class *inappropriate* that is newly introduced to detect

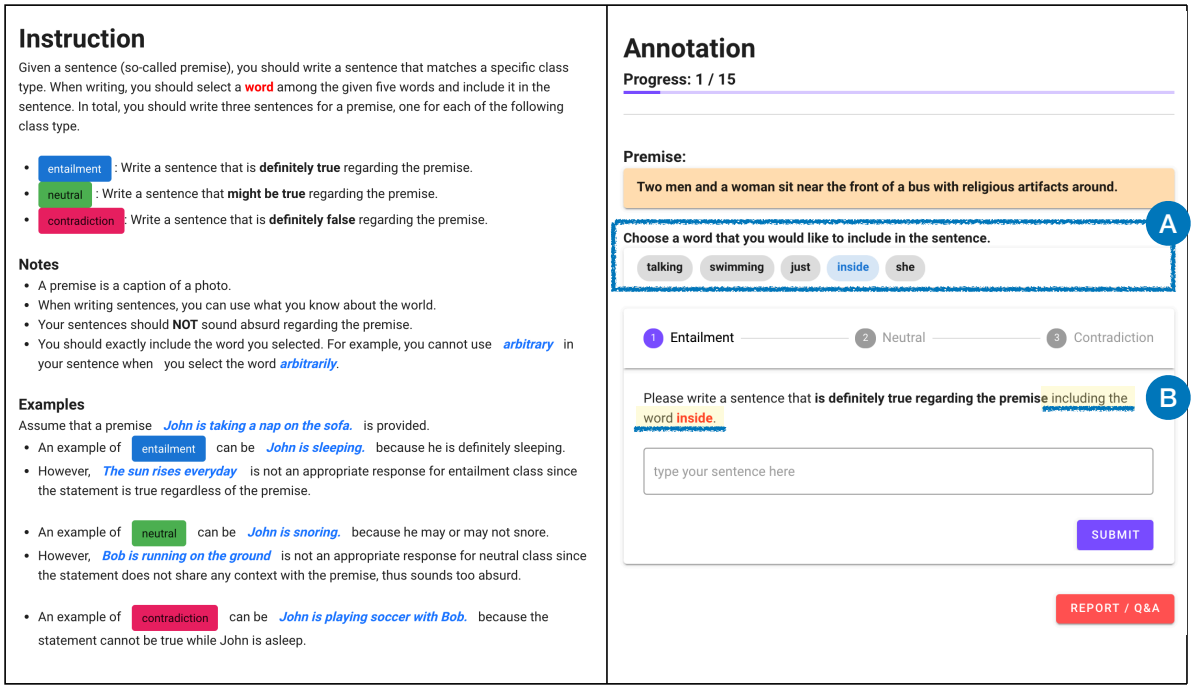

Figure 3.1: The data collection interface used for condition *MW*. The interface of condition *Baseline* does not include counterexamples in the instruction, (A) constraint panel, and (B) a prompt "including the word XXX." The interface of condition *SW* is the same with this interface except (A) the panel.

out-of-context instances. Annotators were prompted to label *inappropriate* when two statements do not share the same context so that the hypothesis sounds absurd regarding the premise. Among the five labels including the label presented to the writer of a hypothesis (author's label), we confirmed the label with three or more counts as gold label, but marked it as *invalid* when no such label exists. Table 3.3 provides the results of validation for the data in each condition, and Table 3.4 shows the number of instances for each class and condition, categorized based on the gold label.

An annotator was allowed to label as many instances as they wish. However, to ensure the reliability and consistency of validation, we asked annotators to label at least 20 instances and added qualification tasks which the workers cannot notice during annotation as they look the same with other annotation tasks. The annotations are ignored when the annotator quit before labeling 20 instances or chose the wrong answer in more than half of the qualification tasks.

We recruited annotators on MTurk with the same qualification requirements in Section 3.2 until all data are labeled by four different annotators. As soon as the number of instances that need one additional label reaches 20, we stopped recruiting workers and annotated the instances by ourselves not seeing the other annotators' labels. We paid \$0.05 per task. In total, 43 annotators labeled about 130 pairs of statements on average including qualification tasks. The median time taken for a single annotation task was 9 seconds.

3.4 Evaluation

To measure the degree of annotation artifact, we define the metric *performance–gap* as follows. In this paper, the performance–gap score measured in terms of accuracy and F1-macro score are referred to as

	SNLI	MNLI	*Baseline*	*SW*	*MW*
Pairs with unanimous gold label	58.3	58.2	39.0	30.1	22.2
Individual label = gold label	89.0	88.7	82.2	79.8	77.1
Individual label = author's label	85.8	85.2	74.0	66.5	64.0
Gold label = author's label	91.2	92.6	88.0	80.1	80.9
Gold label ≠ author's label	6.8	5.6	8.3	11.9	10.1
No gold label (no 3 labels match)	2.0	1.8	3.7	8.0	9.0

Table 3.3: The validation results of data in each condition, including those of the exisitng NLI datasets, SNLI [2] and MNLI [3]. The proportion of individual labels that match the gold label or author's label is calculated with respect to the valid data only, while others are calculated on all data instances.

Gold Label	*Baseline*	*SW*	*MW*
Entailment	210	153	149
Neutral	234	284	262
Contradiction	199	172	189
Inappropriate	7	12	13
No gold label	25	54	68
5 votes	263	203	150
4 votes	233	218	244
3 votes	147	188	200

Table 3.4: The class distribution and the number of majority votes for the data collected in each condition, as a result of data validation. Originally, 225 instances were collected for each class before validation.

accuracy–gap and F1–gap score, respectively.

Definition 2.1. *Performance–gap of A on B*
Given an arbitrary dataset A, we train a hypothesis-only classifier on the training set of A. We then measure the difference between the performance of the trained model and a random classifier on dataset B. We name this metric *performance–gap of A on B* in this paper. This metric is regarded as the degree of annotation artifact present in A which also gives a clue about B. For example, when dataset A has annotation artifact that is also the artifact in B, the performance–gap score would be significantly larger than zero by leveraging the artifact during testing on B. The more severe annotation artifact that two datasets share, the higher the performance–gap score is.

Definition 2.2. *Performance–gap of A*
To measure the absolute level of annotation artifact in a single dataset, we train and test a hypothesis-only classifier on the training and test set of A. For instance, if dataset A has annotation artifact, the performance–gap score of A would be significantly larger than zero as it leverages the artifact during testing.

In the experiment, the dataset obtained from each condition split into train, validation, and test set with the ratio of 7:1:2. When conducting analyses on an existing dataset such as SNLI, we randomly

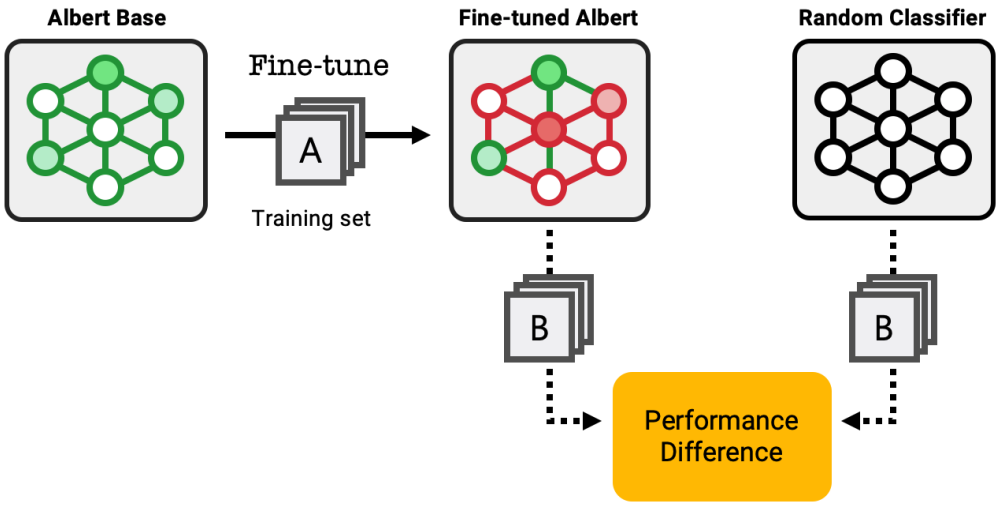

Figure 3.2: The schematic diagram showing the evaluation of performance–gap score of dataset A on dataset B. When measuring the degree of annotation artifact inside dataset A, B is replaced with the test set of A.

sampled 472, 67, and 135 instances from train, validation, and test set, respectively. This adjustment would make the comparison between the conditions fair by keeping the dataset size and the ratio of train, valdiation, and test set in each condition almost identical. The training data was oversampled to balance over classes because through validation, the data instances of either *inappropriate* or no gold label were excluded.

For evaluation, we used a model which consists of a single softmax classification layer on top of ALBERT-Base [29]. The model was fine-tuned for 20 epochs with AdamW optimizer of the learning rate 1e−5, linearly scheduled with warm-up. After fine-tuning, one with the best F1-macro score on the validation set was selected for testing. We measured the accuracy–gap and F1-macro–gap (F1–gap) scores as a result of 100 executions. Since we randomly split the dataset for each execution, the measurement of each execution is independent from others. For statistical testing in the analyses, the two-tailed t-test is used.

3.5 Results

Table 3.5 shows the examples of data instances collected in experiments.

3.5.1 Constraint words

Table 3.6 shows the top five constraint words that are most frequently provided to users during the data collection in condition *SW*. The following analyses explain why the words are selected as constraints. When writing statements of *neutral* class, participant P20 used the word *may* 14 times. This results in the imbalance of the word, which makes the word as a constraint for the two other classes. The usage pattern of constraint words in *neutral* class—such as *there*, *inside*, and *outside*—indicates that users likely describe location or existence of an object in a premise, which is a pattern that SNLI dataset

exhibits as well [6, 7]. Regarding the word *not*, an interesting usage pattern is observed that when a user is given an absurd constraint word regarding a premise, they wrote a statement of *entailment* class by denying the occurrence of an irrelevant object or event. The word *be* is the predominant constraint word in *contradiction* class as it is relatively difficult to write a *contradiction* statement including an auxiliary verb.

3.5.2 Annotation artifacts

The performance gap scores of the dataset collected in each condition are presented in Table 3.7. The accuracy–gap and F1–gap scores of SNLI are ($M = 14.01, SD = 6.61$) and ($M = 13.35, SD = 7.89$), respectively.

***Baseline* vs. SNLI** The difference of both the accuracy–gap ($t(198) = 4.679, p \ll 0.01$) and F1–gap scores ($t(198) = 3.609, p \ll 0.01$) between condition *Baseline* and SNLI dataset is found to be significantly different. This difference can be explained in part by the lower diversity in premises and workers compared to SNLI, which might expose a weakness that annotation artifact made by a single worker brings relatively bigger impact on the data of condition *Baseline* than SNLI.

Baseline* vs. *SW The comparison between two conditions reveal that the annotation artifact was significantly reduced in condition *SW*. The accuracy–gap score of data in condition *SW* is significantly smaller than that of condition *Baseline* ($t(198) = 8.982, p \ll 0.01$), and the same holds for the F1–gap score ($t(198) = 9.652, p \ll 0.01$). Another observation is that the performance–gap scores of condition *Baseline* on *SW* and those of *SW* on *Baseline* are similar and have non-zero values. This indicates that datasets generated in both conditions share a certain amount of annotation artifacts.

Another interesting observation is that the performance–gap scores of condition *Baseline* on *SW* and that of *SW* on *Baseline* are found to be similar. The small difference between the two performance measures is likely to be related to the limitation of a word-level constraint; there can exist diverse levels of annotation artifacts, such as syntactic level, and the word-level intervention to the workflow is not sufficient to fully eliminate the annotation artifact from the dataset. Thus, further research could investigate the diverse factors influencing annotation artifacts.

3.5.3 Task load

Following the addition of a word constraint, however, we find that the task of condition *SW* becomes significantly more difficult than *Baseline*. First, the overall time taken for a user to complete the task significantly increased in condition *SW* compared to the baseline. While participants of condition *Baseline* spent 23m 35s to write all 45 statements, it took 53m 15s for participants of condition *SW* to complete the task on average ($t(28) = -4.080, p \ll 0.01$). In addition, users' feedback on the task collected via the questionnaire supports our claim. Four among 15 participants in condition *SW* left comments that they sometimes felt it was impossible to write a sentence using a constraint word. It is apparent from Table 3.4 that more instances of condition *SW* were judged to be invalid (either *inappropriate* or no gold label) than condition *Baseline*, and that a number of instances originally collected for *entailment* and *neutral* class were categorized to *netural* class. Also, it is observed that the proportion of individual annotators' labels matching with an author's label in condition *SW* is lower than *Baseline*, possibly due to the increased ambiguity in the responses of condition *SW*. These observations imply

that the task of condition *SW* becomes more difficult than *Baseline*, and that the increment of difficulty mostly comes from writing instances of *entailment* and *neutral* class.

Premise	*Hypothesis*	*Annotation*
condition *Baseline*		
Two men push three wheeled chairs up an inclining road.	The two men are using their strength.	(**3**, 2, 0, 0)
A greyhound with a muzzle runs on a racetrack.	A greyhound with a muzzle is exercising.	(**2**, **3**, 0, 0)
A greyhound with a muzzle runs on a racetrack.	The greyhound is running in a race.	(2, **3**, 0, 0)
Two ladies all dressed up and partying on the street.	Ladies are having a good time.	(**3**, **2**, 0, 0)
Two men and a woman sit near the front of a bus with religious artifacts around.	Two men and a woman are driving the bus together.	(1, 0, **3**, 1)
Two football players are tackling an opposing football player with a referee nearby.	A football player escapes tackle and runs down the field to score a touchdown.	(0, **3**, **2**, 0)
condition *SW*		
A greyhound with a muzzle runs on a racetrack.	They muzzled the greyhound that is running on the racetrack.	(**3**, 2, 0, 0)
A greyhound with a muzzle runs on a racetrack.	The greyhound will be waiting for the race to end before it's muzzle is taken off.	(**2**, **3**, 0, 0)
Two men and a woman sit near the front of a bus with religious artifacts around.	There may be no more room on the bus.	(0, **3**, 1, 1)
A woman stands outside of a church door alone.	She is at the front of the church.	(**3**, **2**, 0, 0)
Two football players are tackling an opposing football player with a referee nearby.	The two football players agreed to tackle the referee if he touched the ball.	(0, 1, **3**, 1)
A person staring at a wall that has a bike against it.	The bike may just be an image and not be real	(0, **3**, **2**, 0)
condition *MW*		
Two men dressed up share a toast.	Two men lift their cups with drinks inside.	(**3**, 1, 1, 0)
A blond woman is standing outside a Modell's store with a large tote bag.	The blond woman brought the tote bah with her to Modell's.	(**2**, **3**, 0, 0)
Two men and a woman sit near the front of a bus with religious artifacts around.	They are traveling by roadway.	(1, **3**, 0, 1)
The mom is getting ready to give her baby a bath.	The mom put the baby in the bathtub.	(**3**, **2**, 0, 0)
Two men and a woman sit near the front of a bus with religious artifacts around.	Two men and a woman having a party.	(0, 0, **3**, 2)
Two men push three wheeled chairs up an inclining road.	The men realize the road is a dead end.	(0, **3**, **2**, 0)

Table 3.5: The example data instances where the number of majority label is three for each condition. For half of the examples, the author's label and gold label match while another half do not. Quadruples on the last column indicate the number of received labels of *Entailment*, *Neutral*, *Contradiction*, and *Inappropriate* classes. The red and blue colored labels indicate the **gold label** and the **author's label** for the instance. **Author's label** is marked only when gold label and author's label do not match.

Entailment	#	*Neutral*	#	*Contradiction*	#
may	11	not	17	be	10
car	9	inside	10	may	9
for	9	in	10	motorcycle	7
wearing	8	there	10	outside	7
waiting	8	outside	8	after	6

Table 3.6: The five words that are most frequently used constraints in each condition, during the data collection of condition *SW*. The number on the right side of each word represents the frequency.

(a) Accuracy–gap scores

A \ B	*Baseline*	*SW*	*MW*
Baseline	(18.91, 8.12)	(7.56, 4.70)	(7.09, 4.14)
SW	(6.72, 2.84)	(9.71, 6.23)	(9.98, 3.89)
MW	(7.18, 4.11)	(10.06, 4.92)	(11.06, 7.64)

(b) F1–gap scores

A \ B	*Baseline*	*SW*	*MW*
Baseline	(18.01, 10.22)	(5.81, 6.10)	(5.72, 6.10)
SW	(5.30, 3.64)	(6.33, 6.49)	(7.22, 4.59)
MW	(5.26, 6.06)	(5.39, 6.16)	(8.12, 8.55)

Table 3.7: The degree of annotation artifacts measured by performance–gap scores of A on B. The first and second value indicate the mean and standard deviation of performance–gap scores measured for 100 testings.

Chapter 4. Study 2

While the annotation artifact is significantly reduced following the introduction of a word-level constraint to the crowdsourcing workflow of data collection, another problem is discovered that the task becomes too laborious with the adjustment. In fact, the strict constraint that requires a user to include a specific word in their hypothesis sentences seems to be the major factor for the increased task difficulty. Thus, by slightly easing the constraints, we further explore the design space to investigate the relationship between the degree of annotation artifact, task time, and task design in this section.

4.1 Condition

4.1.1 Multi-word constraint (MW)

To give a less difficult restriction to the users than condition *SW*, we provided five words among which users can select one as their constraint word in condition *MW* (Figure 3.1). Similar to condition *SW*, the five words with the least PMI values are presented as candidate constraint words.

4.2 Data collection and evaluation

The data for condition *MW* was collected in the same way as described in Section 3.2. The participants who had already joined the task of condition *Baseline* and *SW* were not allowed to participate in this task again. For data collection, we paid \$6 (\$9.35/hr) to a participant. For data validation, the reward was \$0.05 per task, and the median time taken for a single annotation task was 8 seconds. In total 23 annotators labeled about 122 pairs of statements on average including qualification tasks.

4.3 Results

4.3.1 Constraint words

Table 4.1 includes the top five constraint words that workers selected the most during the data collection in condition *MW*. Compared to Table 3.6, it seems that workers exhibit the similar pattern when writing a statement of *neutral* class. A constraint word *sleeping* in *entailment* class indicates that workers use this word to make a contradictory description on an active person or animal. Constraint words in *contradiction* class are predominantly used in the hypotheses of *entailment* class in the data of condition *Baseline*, as the premises include those words.

4.3.2 Annotation artifacts

Figure 4.1 briefly depicts the trade-off relationship between the degree of annotation artifact and the task time. As a reference, we included the counterfactually-augmented dataset (CF) [15]. According to their results from the experiment with BiLSTM, annotation artifacts almost vanish in the dataset, while workers spent about four minutes for the unit crowdsourcing task of dataset augmentation. The accuracy–gap and F1–gap scores of the dataset are ($M = 7.03, SD = 6.10$) and ($M = 6.13, SD = 7.00$),

Entailment	#	*Neutral*	#	*Contradiction*	#
in	18	not	10	football	11
with	11	in	9	bike	10
his	10	with	9	motorcycle	8
home	8	outside	8	has	7
sleeping	8	sleeping	8	umbrella	7

Table 4.1: The five constraint words that workers selected the most in each condition, during the data collection of condition *MW*. The number on the right side of each word represents the frequency.

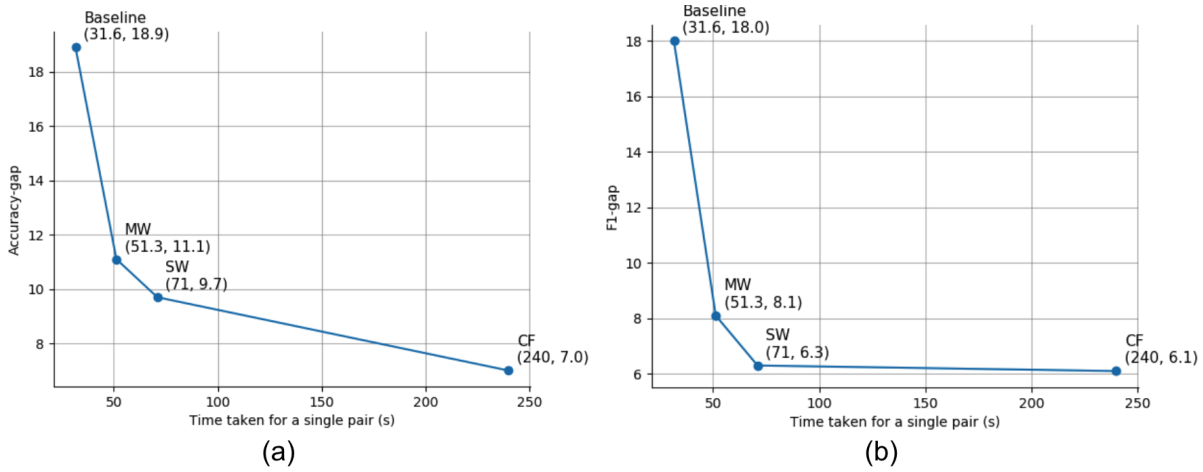

Figure 4.1: Two graphs showing the relationship between the time taken for a unit task and the degree of annotation artifact measured by (a) accuracy–gap and (b) F1–gap.

respectively. Thus, we consider the experiment result of this dataset as one extreme of the design space. From Figure 4.1, we can interpret that while the counterfactually-augmented data [15] succeeds at reducing the annotation artifact, the task design is too inefficient in terms of unit task time. The accuracy–gap ($t(198) = -1.365, p = 0.174$) and F1–gap ($t(198) = -1.669, p = 0.097$) scores are both increasing in condition *MW* with giving higher freedom in writing to users than condition *SW*, while the increment of accuracy–gap is not statistically significant. The observed increase of annotation artifact in condition *MW* than *SW* is possibly attributed to the selection bias, the workers' strategies to choose a word that is similar to the premise sentences, considering that the set of premises provided to users are identical.

4.3.3 Task load

The time taken for completing the task of condition *MW* is 38 minutes 30 seconds on average, which is significantly more than the task time of condition *Baseline* ($t(28) = -2.6207, p = 0.014$), while less compared to condition *SW* ($t(28) = 2.043, p = 0.051$). Also, two of 15 users explicitly left comments that they felt the task demanding. To sum up, as we expected with providing multiple options, users may find this task easier than the task of condition *SW*.

Chapter 5. Discussion

In this section, we discuss how annotation artifacts could be attributed to word patterns. Then, we discuss the trade-off between work load and dataset quality by controlling the writing constraints. Furthermore, we suggest design considerations for crowdsourcing workflows for data generation.

Annotation artifacts can be attributed to word patterns. The results of Study 1 (Table 3.7) reveal that annotation artifacts are significantly reduced in condition *SW* than condition *Baseline*, following the introduction of a word constraint which shows the most skewed distribution over classes. The possible explanation for this result is that the word-level constraints successfully prevented users from leveraging the word-level strategies, which signals that the generation of annotation artifacts is partly attributed to the word patterns.

Controlling the writing constraints leads to trade-off between task load and dataset quality. The results of Study 2 (Figure 4.1) reveal the trade-off that easing the writing constraints decreases the task load at the cost of an increase in annotation artifacts present in the dataset. However, controlling the degree of freedom in writing in a more sophisticated manner could lead to a more optimized solution in terms of dataset quality and task load. The participants in condition *SW* explicitly mentioned that writing an entailment is especially difficult compared to writing the other two classes. Also, the validation result (Table 3.4) indicates that many responses originally written for *entailment* class were incorrectly calssified into other classes. Based on these observations, we can have the *SW* constraints in other classes while applying *MW* constraints only for the *entailment* class. As such, we can achieve a more optimized task design with more sophisticated control of writing constraints.

5.1 Design considerations for data generation

The results in Study 2 (Figure 4.1) suggest the possibility of designing a data collection task to reduce the annotation artifact while workers are continuously incentivized to contribute. However, to examine the impact of an individual factor in annotation artifacts, the experiments were conducted in a controlled setting so that only a slight variation of a single factor was introduced between conditions. However, when a dataset is collected in practice, more diverse degrees of variations could be introduced on the task design in terms of objective functions, degree of freedom, and constraint type.

Objective function In this research, a constraint word is selected based on the PMI value which measures the extent to how skewed the usage of a word is over classes. We chose this metric as we believed that the distorted distribution of a word can act as an indicator of annotation artifact. Assuming a dataset designer puts emphasis on the diversity in the dataset among other things, one can try another objective function to choose a word that is located far from the sentence on the embedding space. As dataset construction inevitably includes value-laden decisions, investigating different objective functions and their impact could present valuable insights.

Degree of freedom We can control the degree of freedom in more diverse dimensions than the number of options, such as financial compensation, time, etc. For example, from the task design of condition *MW*, a designer may not want to harm the degree of freedom while hoping the annotator to choose a specific word from the provided options. Although this is almost impossible in our experiment setting, one can adjust the task design in practice by distinguishing the amount of monetary reward according to the preference and needs.

Constraint type While we chose to focus on the word-level constraint regarding its applicability over various domains, another type of constraints such as syntactic patterns can be considered. Considering the previous study that a neural model of NLI exhibits poor performance on several syntactic heuristics, we speculate that a certain type of syntactic patterns in the dataset can be attributed to annotation artifacts. As such, considering the diverse candidates that possibly affects the generation of the annotation artifact, our study design can be adopted to investigate the role of particular factors by adjusting the type of constraint.

5.2 Limitations

The current study has several limitations that need to be taken into account when generalizing the reported findings to other settings.

5.2.1 Task design

Despite the success at mitigation of artifacts with the introduction of word-level constraint, the task design has a weakness that the PMI value is updated in real time once a user submits a response. However, since the responses that are invalid or do not follow the instructions cannot be filtered out along with the data collection, a concern arises that the constraint word can be selected from the word distribution contaminated with invalid responses. In an extreme case of when this keeps happening, an irrelevant constraint word may be suggested to users, which can ultimately harm the quality of dataset. The post-hoc analyses of our experiments revealed that the number of invalid responses submitted to the system is negelectable with respect to the number of valid instances. However, when introducing a real-time constraint to the workflow design, the impact of invalid responses to the selection of a constraint should be carefully considered.

5.2.2 Payment

The average hourly wages that workers are paid in each condition were \$12.57/hr, \$6.76/hr, and \$9.35/hr, for condition *Baseline*, *SW*, and *MW*, respectively. This raises a concern that the unfair treatment between conditions might affect the experiment result. While acknolwledging the concern, the lower payment likely leads workers to leverage more strategies, thus a higher degree of annotation artifacts is expected than we pay the fair amount. Nontheless, the annotation artifacts are found to be significantly lower in condition *SW* and *MW*, and we argue that this issue does not harm the integrity of our interpretations. However, in the ethical perspective, we failed to ensure the U.S. federal minimum wage (\$7.25/hr as of 2020) to the participants in condition *SW*, despite our efforts to correctly estimate the task time through a lab study prior to the MTurk study.

Chapter 6. Conclusion

In this research, to reduce annotation artifact, we introduce word-level constraints into the crowdsourcing workflow for data collection. Given the single-word constraint during NLI data collection, annotators generate a dataset with significantly reduced annotation artifact. This observation suggests that the word pattern in the data collected without any interruption can be leveraged for early-stopping the proliferation of annotation artifacts during data collection with an appropriate intervention. However, the introduction of constraint makes the data generation task immensely difficult, which reduces the practical value of the method. Thus, to further understand the relationship between the degree of annotation artifact and task design, we collected another set of data allowing annotators to choose a constraint word among several options. The results reveal that there is a trade-off between annotation artifact and freedom in writing, and this trade-off can be a topic for future research. Based on the analyses, we propose a possible improvements of the workflow design which the dataset designers may find it helpful when designing the crowdsourcing workflow for collecting NLP dataset with writing.

Bibliography

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, November 2016. Association for Computational Linguistics.

[2] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 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, September 2015. Association for Computational Linguistics.

[3] Adina Williams, Nikita Nangia, and Samuel Bowman. 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, June 2018. Association for Computational Linguistics.

[4] Fabrizio Morbini, Eric Forbell, and Kenji Sagae. Improving classification-based natural language understanding with non-expert annotation. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 69–73, 2014.

[5] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. Story cloze task: Uw nlp system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 52–55, 2017.

[6] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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, June 2018. Association for Computational Linguistics.

[7] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 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, June 2018. Association for Computational Linguistics.

[8] Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[9] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. ACL, 2020.

[10] Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 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, June 2019. Association for Computational Linguistics.

[11] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 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, November 2019. Association for Computational Linguistics.

[12] Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, 2019.

[13] He He, Sheng Zha, and Haohan Wang. 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, November 2019. Association for Computational Linguistics.

[14] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 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, July 2020. Association for Computational Linguistics.

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

[16] Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, 2018.

[17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[18] Zheng Cai, Lifu Tu, and Kevin Gimpel. Pay attention to the ending: Strong neural baselines for the roc story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 616–622, 2017.

[19] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[20] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, 2016.

[21] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in neural information processing systems*, pages 841–852, 2019.

[22] Samuel R. Bowman, Jennimaria Palomaki, Livio Baldini Soares, and Emily Pitler. New protocols and negative results for textual entailment data collection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8203–8214, Online, November 2020. Association for Computational Linguistics.

[23] Stefan Larson, Anthony Zheng, Anish Mahendran, Rishi Tekriwal, Adrian Cheung, Eric Guldan, Kevin Leach, and Jonathan K. Kummerfeld. Iterative feature mining for constraint-based data collection to increase data diversity and model robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8097–8106, Online, November 2020. Association for Computational Linguistics.

[24] Herbert A Simon. Rational choice and the structure of the environment. *Psychological review*, 63(2):129, 1956.

[25] Daniel M Oppenheimer, Tom Meyvis, and Nicolas Davidenko. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of experimental social psychology*, 45(4):867–872, 2009.

[26] Adam Kapelner and Dana Chandler. Preventing satisficing in online surveys. *Proceedings of Crowd-Conf*, 2010.

[27] Tyler Hamby and Wyn Taylor. Survey satisficing inflates reliability and validity measures: An experimental comparison of college and amazon mechanical turk samples. *Educational and Psychological Measurement*, 76(6):912–932, 2016.

[28] Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China, November 2019. Association for Computational Linguistics.

[29] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

Acknowledgment

2년 간 많은 도움을 주셨던 오혜연 교수님과 김주호 교수님께 감사드립니다. 가족과 연구실 동료, 그리고 이 글을 읽는 여러분 덕분에 무사히 끝을 맺습니다. 고맙습니다.