Reproducibility Report

Andressa Chan, Millicent Li, Mino Nakura
University of Washington
Seattle, Washington
{ac426, limill01, nakuram}@uw.edu

Reproducibility Summary

Scope of Reproducibility

The motivation of this paper is to introduce a dataset so models can interpret indirect answers on a more granular level. Therefore, we want to verify the performance of this sequence classification task on this new dataset, Circa. Specifically, the paper claims to achieve around 75-85% accuracy using various training methods.

These are the hypotheses we are testing in this work:

- Claim 1: Finetuning BERT on Circa using both matched and unmatched training can achieve the same accuracies within a margin of +/- 5% for the strict definition of labeling, which contains 9 classes.
- Claim 2: Finetuning BERT on Circa using both matched and unmatched training can achieve the same accuracies within a margin of +/- 5% for the relaxed definition of labeling, which contains 6 classes.
- Claim 3: Pretraining the BERT models on different datasets (BOOLQ, MNLI, DIS) and then training on Circa can achieve higher accuracy at classifying on the Circa dataset compared to training only on Circa.

Methodology

We focused on reproducing the results from scratch. In particular, we used existing libraries to reimplement the models. The overall time taken to produce the results was close to 70 hours of model running time, running on a mix of Azure Titan K80 and GTX 1080 Ti GPUs.

Results

We report our development and test accuracies, as well as our F scores in a detailed table for the strict and relaxed versions of the dataset. Overall, our work does not support claims 1 and 2 made by the paper, and the test accuracies we achieve are not within a +/-5% margin of the paper's. However, for claim 3, our results prove our hypothesis.

What was easy

We benefited from the authors' use of a commonly used model in BERT, which allowed us to use the HuggingFace library. Furthermore, the author relied on publicly available datasets to conduct their experiments, which made obtaining this information and cleaning the data for training and validating simple.

What was difficult

The difficulty of reproducing this paper stemmed from the number of experiments conducted. Overall, there were 8 different models we had to train. For each model trained on Circa, we had two experimental settings. For all combinations, we had to train a total of 134. Training and testing all of these models took a long time, although it was feasible across a few days and split among multiple GPUs.

Communication with original authors

We communicated with the authors concerning the rewritten version of the Circa dataset, which could be obtained publicly through the repository listed in the paper.

CSE 517 2021, Reproducibility Project Report.

1 Introduction

In everyday conversation, humans give complex responses to simple “yes/no” questions. These responses are indirect, and interpreting the sentiments behind them are often difficult for machines. In the paper “‘I’d rather just go to bed’: Understanding Indirect Answers”, the authors attempt to tackle this problem by curating a new large scale dataset called Circa. Specifically, the paper publishes a dataset that can classify indirect answers into conditionals, uncertain, and middle-ground answers, which are more diverse classifications than existing work. Using this new dataset, they train a combination of multiple baseline models and models pre-trained on Circa and are able to see an improvement on the interpretation of indirect answers.

2 Scope of reproducibility

The motivation of this paper is to introduce a dataset so models can interpret indirect answers on a more granular level (i.e. more categories for classification). Therefore, we want to verify the performance of this sequence classification task on this new dataset, Circa. Specifically, the paper claims to achieve around 75-85% accuracy using various training methods. There are two experimental settings: matched and unmatched. The matched setting refers to when all ten question scenarios are present in the training and testing data, while the unmatched setting withholds one question scenario from the training data and uses it for testing.

These are the hypotheses we are testing in this work:

- Claim 1: Finetuning BERT on Circa using both matched and unmatched training can achieve the same accuracies within a margin of +/- 5% for the strict definition of labeling, which contains 9 classes.
- Claim 2: Finetuning BERT on Circa using both matched and unmatched training can achieve the same accuracies within a margin of +/- 5% for the relaxed definition of labeling, which contains 6 classes.
- Claim 3: Pretraining the BERT models on different datasets (BOOLQ, MNLI, DIS) and then training on Circa can achieve higher accuracy at classifying on the Circa dataset compared to training only on Circa.

In claim 1, there are 9 main classes of labels that the models use in the sequence classification task: Yes; No; Probably yes / sometimes yes; Yes, subject to some conditions; Other; Probably no; In the middle, neither yes nor no; I am not sure how X will interpret Y’s answer; and N/A. These are the more strict definitions for labels. We are verifying whether their reported results in the paper can be achieved by training BERT on the Circa dataset with strict labels on both the matched and unmatched scenario cases.

In claim 2, the 6 classes of labels is a relaxed version of the 9 classes: Yes; No; Yes, subject to some conditions; In the middle, neither yes nor no; Other; and N/A. Compared to the six classes, ‘probably yes / sometimes yes’ is relaxed to ‘yes’ and ‘probably no’ becomes ‘no’. We are verifying whether their reported results in the paper can be achieved. The authors train a BERT model on the relaxed Circa dataset and provide statistics on the performance their trained models can achieve. To verify this claim, we run similar experiments to verify that we can replicate their results.

In claim 3, the paper presents various models that are pretrained on different datasets to show that accuracy performance can be improved on the Circa dataset. Here, we are trying to verify the different accuracy results they achieve. To verify this claim, we are focusing on pretraining on BOOLQ, MNLI, and DIS, establishing a baseline with BOOLQ and MNLI, then further finetuning on these Circa before finally evaluating on the Circa dataset.

Claims 1 and 2 are the central claims in the paper. The goal of the paper is to introduce the new Circa dataset and its statistics after training BERT on it, with the intent to show that significant improvements have been made to the task of understanding indirect answers. Claim 3 is not a central claim but further claims that there could be more improvements in accuracy on finetuning on the Circa dataset by pretraining on other datasets, such as BOOLQ and MNLI, that focus on classifying sentences.

3 Methodology

3.1 Model descriptions

In the paper, the authors use BERT, a transformer, as their model (Devlin et al. 2018). This model takes in a sequence of tokens as input, and outputs a label from a predefined number of classes. The learning objective is to minimize the classification loss. The pre-trained BERT model has 12 layers, 768 hidden units, 12 attention heads, and 110M parameters. The same model is used across all experiments in the paper. Finetuning was done on a single Cloud TPU with a total finetuning time of 30 minutes.

In our reproduced version, we also focused our experimentation on BERT, using Huggingface's Transformers library (Wolf et al. 2019) for the implementation. Specifically, we used BertForSequenceClassification with the same parameters as the paper and used the same training objective. The only difference was that we trained the models on a mix of Azure Virtual Machines (NVIDIA Tesla K80) and lab machines (NVIDIA GTX 1080 Ti).

3.2 Datasets

Dataset Name	CIRCA (unmatched)	CIRCA (matched)	BOOLQ	MNLI	DIS	SNLI
Number of Total Examples	∼34260	∼34260	15800	410000	560827	410000
Number of Training Examples	∼27700	∼20561	9400	392000	400000	392000
Number of Development Examples	∼3080	∼6855	3200	9000	160827	9000
Number of Testing Examples	∼3480	∼6855	3200	9000	-	9000

*These splits were approximate for each dataset. For instance, depending on whether we were training on a matched or unmatched case, the splits would differ depending on the case. Thus, these are simply approximations for the CIRCA dataset

We obtained three datasets – Circa, BOOLQ, and DIS – all of which are publicly available on GitHub. The Circa dataset is available as a Tab Separated Value, where each row consists of a scenario, question, indirect answer, and a label which classifies the answer. The Circa dataset is curated by crowd sourcing yes/no questions from annotators based on dialogue scenarios. Then, a different set of annotators is asked to provide indirect answers to the questions. Lastly, five annotators judge the label for the indirect answer, which is then used as the gold standard label for the question and scenario.

Circa. This dataset contains 34,268 question-answer pairs, with each question-answer pair associated to a specific context, also known as a scenario. There are a total of ten different scenarios in the dataset. Furthermore, each question-answer pair also has two labels that interpret the indirect answers as a direct response, one for the strict labels and another for the relaxed labels. Each label is determined by a committee of five annotators that judge how the indirect answer should be interpreted.

BOOLQ. The BOOLQ corpus (Clark et al. 2019) is a reading comprehension question-answering dataset that focuses on naturally occurring yes/no questions. These are web queries paired with Wikipedia paragraphs that help answer the question. There are two target labels, yes and no, for the responses. In total, there are 9.4k training examples, 3.2k development examples, and 3.2k test examples.

DIS. The DIS corpus (Nie et al. 2019) is a dataset used to understand the discourse relationships between two sentences. There are 5 different target labels, and draws 4,706,202 pairs of sentences as training examples.

MNLI. The MultiNLI corpus (Williams et al., 2018) is a dataset for textual entailment. Each data point is made up of a premise-hypothesis pair, which is classified into one of three classes: 'entailment' 'contradiction', and 'neutral'.

For BOOLQ, DIS, and and MNLI, the corpora come with predefined splits for the training and validation data. The data can be obtained online in their public repository for each of the corpus. On the other hand, Circa requires different splits for two matched and unmatched experimental cases, matched and unmatched.

There are two different experimental settings for Circa: matched and unmatched. For the matched scenario, the dataset is split into a 60-20-20 train-test-dev. For the unmatched case, we generate 10 total different datasets, where each dataset excludes one out of the ten scenarios from the train set and dev set and leaves it for the test set. Therefore, in the unmatched case, the split roughly becomes 80-10-10 for each of the scenarios. For both the matched and unmatched cases, we manually split the dataset ourselves.

3.3 Hyperparameters

The paper mentions three hyperparameters that were tuned in their experiments. The three hyperparameters are the learning rate, number of epochs, and the batch size. The base models (MNLI, BOOLQ, DIS) try an exhaustive combination of the following parameters: learning rate (5e-5, 3e-5, 2e-5), number of epochs (2, 3, 4), and train batch size (16, 32). When finetuning on Circa, the parameters are 3 epochs and a batch size of 32 (the learning rates differ per model). For the purposes of our experiments, we will use the hyperparameter combination that produced the best result as listed in the paper. These hyperparameters are shown in the tables below.

Model	Learning rate	Number of epochs	batch size
MNLI	2e-5	3	16
BOOLQ	3e-5	4	16
DIS	2e-5	2	32

3.4 Implementation

This paper largely focuses on the curation of the Circa dataset and does not contain any existing code. We are implementing the dataloaders (for Circa, BOOLQ, MLNI, and DIS) and the training script from scratch in Python. To reproduce the claims in the paper, we rely on publicly available libraries to implement the models and use the mentioned datasets. The model and tokenizer used are from the HuggingFace BERT implementations.

3.4.1 Dependencies

We contain a nlp.yml file in our Github Repo that creates the conda environment necessary to run our code. Some of the dependencies that we used include Huggingface's BERT and BART Transformers, Pandas, tqdm, Pytorch, and others.

3.4.2 Data Download Instructions

All datasets are publicly available and can be accessed online [1][2][3][4]. The Circa and BoolQ Dataset are both available on Google Research Datasets' Github Repository. MNLI is available on the author's website, while the DIS dataset is also available on the author's github. Store each downloaded dataset in the data/ folder of the repository if it is not already present.

3.4.3 Preprocessing Code and Commands

We provide two python scripts to preprocess the Circa Datasets. `datasplit.py` contains code to split the Circa dataset according to the 60-20-20 split necessary for the training, dev, and test for the matched setting. Similarly, the `datasetunmatch.py` will split the data accordingly for the unmatched setting for all 10 scenarios. To run either python script, run `python datasplit.py` or `python dataunmatch.py`. Any other preprocessing is done within the Dataloader, which is called dataloader.py. Since this processing occurs at runtime, we do not have a command to run `dataloader.py` and we directly package the preprocessing with the training.

3.4.4 Training Code and Commands

The model training is in the file `finetuning.py` inside our Github repo. Since there are over 200 different models we have to train, we created bash scripts to run different experimental settings. The bash scripts we ran are inside the /bash_scripts/ folder and have names that start with "finetune_on". Once the models are trained, they are saved in the /models/ directory so we can evaluate them on the testing datasets.

3.4.5 Evaluation Code and Commands

The evaluation is done in the file `testing.py` and `MNLItesting.py`. To run the evaluation, there are a number of bash scripts that are used that start with the prefix test_*:

- test_baseline_boolq_matched_strict.sh
- test_baseline_mnli_matched_strict.sh
- test_dis_circa_matched_relaxed.sh
- ...

and continue with the remaining files that have similar prefixes After running the evaluation script, the results are outputted directly to the screen. Also, the results are saved in .csv files in the /results/ folder where the associated name of the model can be seen with the corresponding results.

[1] https://github.com/google-research-datasets/circa

[2] https://github.com/google-research-datasets/boolean-questions

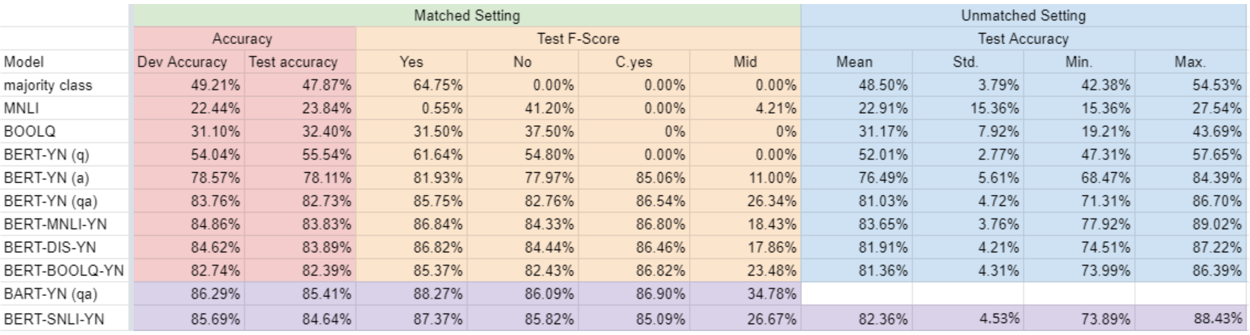
[3] https://cims.nyu.edu/ sbowman/multinli/

[4] https://github.com/windweller/DisExtract

3.4.6 Pretrained model (If Applicable)

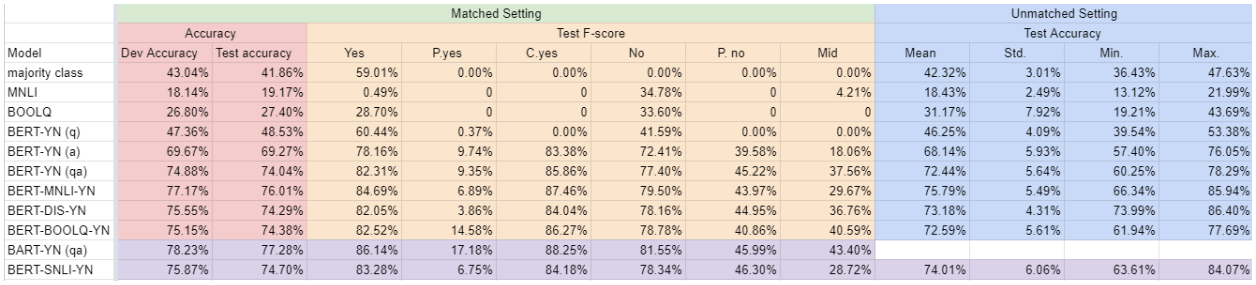
The pretrained model that we used was BertForSequenceClassification from Huggingface's Transformers.

3.4.7 Table of Results



	Matched Setting						Unmatched Setting			
	Accuracy		Test F-Score				Test Accuracy			
Model	Dev Accuracy	Test accuracy	Yes	No	C.yes	Mid	Mean	Std.	Min.	Max.
majority class	49.21%	47.87%	64.75%	0.00%	0.00%	0.00%	48.50%	3.79%	42.38%	54.53%
MNLI	22.44%	23.84%	0.55%	41.20%	0.00%	4.21%	22.91%	15.36%	15.36%	27.54%
BOOLQ	31.10%	32.40%	31.50%	37.50%	0%	0%	31.17%	7.92%	19.21%	43.69%
BERT-YN (q)	54.04%	55.54%	61.64%	54.80%	0.00%	0.00%	52.01%	2.77%	47.31%	57.65%
BERT-YN (a)	78.57%	78.11%	81.93%	77.97%	85.06%	11.00%	76.49%	5.61%	68.47%	84.39%
BERT-YN (qa)	83.76%	82.73%	85.75%	82.76%	86.54%	26.34%	81.03%	4.72%	71.31%	86.70%
BERT-MNLI-YN	84.86%	83.83%	86.84%	84.33%	86.80%	18.43%	83.65%	3.76%	77.92%	89.02%
BERT-DIS-YN	84.62%	83.89%	86.82%	84.44%	86.46%	17.86%	81.91%	4.21%	74.51%	87.22%
BERT-BOOLQ-YN	82.74%	82.39%	85.37%	82.43%	86.82%	23.48%	81.36%	4.31%	73.99%	86.39%
BART-YN (qa)	86.29%	85.41%	88.27%	86.09%	86.90%	34.78%				
BERT-SNLI-YN	85.69%	84.64%	87.37%	85.82%	85.09%	26.67%	82.36%	4.53%	73.89%	88.43%

Figure 1: Statistics for the Relaxed Setting



	Matched Setting								Unmatched Setting			
	Accuracy		Test F-score						Test Accuracy			
Model	Dev Accuracy	Test accuracy	Yes	P.yes	C.yes	No	P. no	Mid	Mean	Std.	Min.	Max.
majority class	43.04%	41.86%	59.01%	0.00%	0.00%	0.00%	0.00%	0.00%	42.32%	3.01%	36.43%	47.63%
MNLI	18.14%	19.17%	0.49%	0	0	34.78%	0	4.21%	18.43%	2.49%	13.12%	21.99%
BOOLQ	26.80%	27.40%	28.70%	0	0	33.60%	0	0	31.17%	7.92%	19.21%	43.69%
BERT-YN (q)	47.36%	48.53%	60.44%	0.37%	0.00%	41.59%	0.00%	0.00%	46.25%	4.09%	39.54%	53.38%
BERT-YN (a)	69.67%	69.27%	78.16%	9.74%	83.38%	72.41%	39.58%	18.06%	68.14%	5.93%	57.40%	76.05%
BERT-YN (qa)	74.88%	74.04%	82.31%	9.35%	85.86%	77.40%	45.22%	37.56%	72.44%	5.64%	60.25%	78.29%
BERT-MNLI-YN	77.17%	76.01%	84.69%	6.89%	87.46%	79.50%	43.97%	29.67%	75.79%	5.49%	66.34%	85.94%
BERT-DIS-YN	75.55%	74.29%	82.05%	3.86%	84.04%	78.16%	44.95%	36.76%	73.18%	4.31%	73.99%	86.40%
BERT-BOOLQ-YN	75.15%	74.38%	82.52%	14.58%	86.27%	78.78%	40.86%	40.59%	72.59%	5.61%	61.94%	77.69%
BART-YN (qa)	78.23%	77.28%	86.14%	17.18%	88.25%	81.55%	45.99%	43.40%				
BERT-SNLI-YN	75.87%	74.70%	83.28%	6.75%	84.18%	78.34%	46.30%	28.72%	74.01%	6.06%	63.61%	84.07%

Figure 2: Statistics for the Strict Setting

3.5 Experimental setup

We ran our experiments on a GeForce GTX 1080 Ti and on Azure Virtual Machine with Tesla K80 GPUs. We use a GPU instead of a TPU (as done in the paper) because we have GPUs readily available, which were far quicker and easier to set up. In addition, we ran our experiments using Bash and Python scripts instead of on a Google Colab or Jupyter Notebook file.

Github link: https://github.com/NakuraMino/IndirectAnswers

3.6 Computational requirements

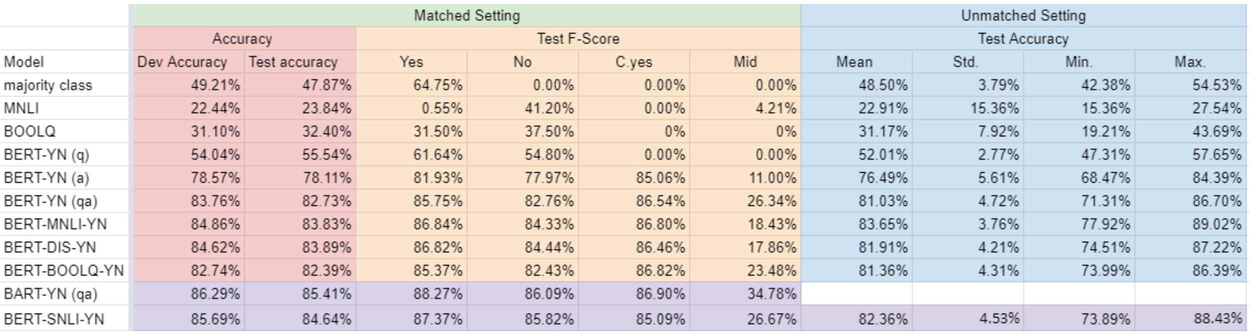
There are a total of three different training regimes we will be experimenting with: answer only, question only, and question-answer (QA) pair models. The answer only and question only will only be trained on Circa using BERT. Within the question-answer pair models, we will be training on four different types of datasets (Circa dataset is referred to as YN): BERT-YN, BERT-BOOLQ-YN, BERT-MNLI-YN, BERT-DIS-YN. Each experiment will take around 30 minutes to complete on a GPU, and the memory required is about 3GB per run.

	Type of hardware	GPU hours used
Expected	Cloud TPU	0.5 hours
Actual	GTX 1080 Ti and Azure VM	66 hours

Model	Circa	MNLI	DIS	BOOLQ
Expected average runtime for each epoch	10 minutes	10 minutes	10 minutes	10 minutes
Actual average runtime for each epoch	10 minute	1 hour	30 minutes	20 minutes
Expected total number of trials	12	8	8	8
Actual total number of trials	66	22	22	22

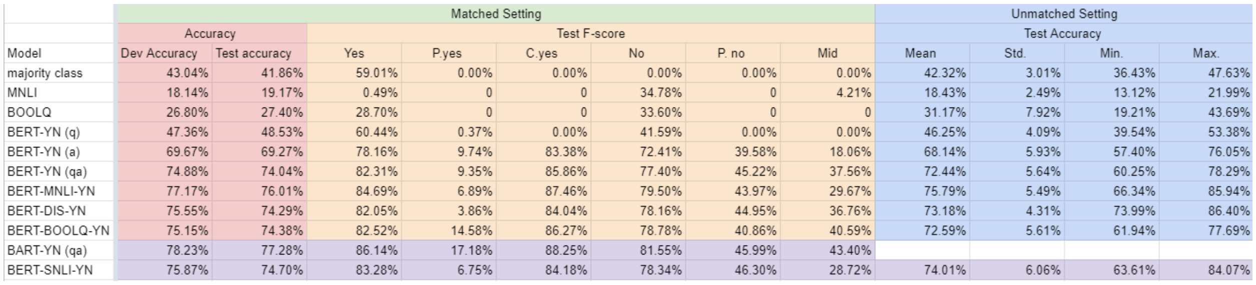
4 Results

Like the original paper, we report our development and test accuracies, as well as our F scores in a detailed table for the strict and relaxed versions of the dataset. Overall, our work does not support claims 1 and 2 made by the paper, and the test accuracies we achieve are not within a +/-5% margin of the paper's. However, for claim 3, our results prove our hypothesis. Our results will be discussed in depth in the sections that follow.



	Matched Setting						Unmatched Setting			
	Accuracy		Test F-Score				Test Accuracy			
Model	Dev Accuracy	Test accuracy	Yes	No	C.yes	Mid	Mean	Std.	Min.	Max.
majority class	49.21%	47.87%	64.75%	0.00%	0.00%	0.00%	48.50%	3.79%	42.38%	54.53%
MNLI	22.44%	23.84%	0.55%	41.20%	0.00%	4.21%	22.91%	15.36%	15.36%	27.54%
BOOLQ	31.10%	32.40%	31.50%	37.50%	0%	0%	31.17%	7.92%	19.21%	43.69%
BERT-YN (q)	54.04%	55.54%	61.64%	54.80%	0.00%	0.00%	52.01%	2.77%	47.31%	57.65%
BERT-YN (a)	78.57%	78.11%	81.93%	77.97%	85.06%	11.00%	76.49%	5.61%	68.47%	84.39%
BERT-YN (qa)	83.76%	82.73%	85.75%	82.76%	86.54%	26.34%	81.03%	4.72%	71.31%	86.70%
BERT-MNLI-YN	84.86%	83.83%	86.84%	84.33%	86.80%	18.43%	83.65%	3.76%	77.92%	89.02%
BERT-DIS-YN	84.62%	83.89%	86.82%	84.44%	86.46%	17.86%	81.91%	4.21%	74.51%	87.22%
BERT-BOOLQ-YN	82.74%	82.39%	85.37%	82.43%	86.82%	23.48%	81.36%	4.31%	73.99%	86.39%
BART-YN (qa)	86.29%	85.41%	88.27%	86.09%	86.90%	34.78%				
BERT-SNLI-YN	85.69%	84.64%	87.37%	85.82%	85.09%	26.67%	82.36%	4.53%	73.89%	88.43%

Figure 3: Statistics for the Relaxed Setting



	Matched Setting								Unmatched Setting			
	Accuracy		Test F-score						Test Accuracy			
Model	Dev Accuracy	Test accuracy	Yes	P.yes	C.yes	No	P. no	Mid	Mean	Std.	Min.	Max.
majority class	43.04%	41.86%	59.01%	0.00%	0.00%	0.00%	0.00%	0.00%	42.32%	3.01%	36.43%	47.63%
MNLI	18.14%	19.17%	0.49%	0	0	34.78%	0	4.21%	18.43%	2.49%	13.12%	21.99%
BOOLQ	26.80%	27.40%	28.70%	0	0	33.60%	0	0	31.17%	7.92%	19.21%	43.69%
BERT-YN (q)	47.36%	48.53%	60.44%	0.37%	0.00%	41.59%	0.00%	0.00%	46.25%	4.09%	39.54%	53.38%
BERT-YN (a)	69.67%	69.27%	78.16%	9.74%	83.38%	72.41%	39.58%	18.06%	68.14%	5.93%	57.40%	76.05%
BERT-YN (qa)	74.88%	74.04%	82.31%	9.35%	85.86%	77.40%	45.22%	37.56%	72.44%	5.64%	60.25%	78.29%
BERT-MNLI-YN	77.17%	76.01%	84.69%	6.89%	87.46%	79.50%	43.97%	29.67%	75.79%	5.49%	66.34%	85.94%
BERT-DIS-YN	75.55%	74.29%	82.05%	3.86%	84.04%	78.16%	44.95%	36.76%	73.18%	4.31%	73.99%	86.40%
BERT-BOOLQ-YN	75.15%	74.38%	82.52%	14.58%	86.27%	78.78%	40.86%	40.59%	72.59%	5.61%	61.94%	77.69%
BART-YN (qa)	78.23%	77.28%	86.14%	17.18%	88.25%	81.55%	45.99%	43.40%				
BERT-SNLI-YN	75.87%	74.70%	83.28%	6.75%	84.18%	78.34%	46.30%	28.72%	74.01%	6.06%	63.61%	84.07%

Figure 4: Statistics for the Strict Setting

4.1 Claim 1: Disproved

Our results did not match our hypothesis of BERT finetuned on Circa achieving accuracies +/-5% of the strict definition of labeling. For both the matched and unmatched cases, BERT trained on Circa (question) was able to reach a +/-5% margin of the paper's test accuracy, with the paper's accuracy being 52.8% and 49.4% for matched and unmatched, respectively, and our accuracies being 48.53% and 46.25%. However, for both BERT trained on Circa (answer) and Circa (question and answer), the test accuracies for both matched and unmatched cases disproves the hypothesis – 77.8% vs. 69.27% for Circa (answer) matched, 75.8% vs. 68.14% for Circa (answer) unmatched, 84.0% vs. 74.04% for Circa (question answer) matched, and 81.2% vs. 72.44 for Circa (question and answer) unmatched.

4.2 Claim 2: Disproved

The result table and experimental results show that the models trained on the Circa dataset does hit similar accuracies within a margin of +/- 5% for the relaxed definition, but some models did not meet the margin of +/- 5%. Therefore, the hypothesis was disproved in this case. The model that did not meet this threshold was the BERT-YN model which had an accuracy of 82.73% for our training and an accuracy of 87.8% for their training, which is a margin of error greater than 5%. The best performing model (the model that had the smallest percentage gap) that we trained in comparison to the model from the paper was BERT-DIS-YN, which had a test accuracy of 83.89% for our testing as compared to a test accuracy of 87.4% from what the paper had. Comparing the remainder of our results to theirs, the margin of error between the other models trained on the Circa dataset was within a margin of error of +/- 5%.

4.3 Claim 3: Proved

From our experiments, we can observe that pretraining on other datasets improves the classification accuracy on the Circa dataset, thus proving the paper's claim. When using strict labels with the Circa dataset, we see that pretraining on different datasets strictly improves model performance. Specifically, the test accuracy score on the BERT model only trained on Circa is 74.04%. On the other hand, the lowest test accuracy score for the BERT models pretrained on another dataset has a score of 74.29% accuracy. When looking at the relaxed labels, we see a similar performance improvement in test accuracies. Specifically, the BERT model pretrained on other datasets reach accuracy scores up to 83.89%, while accuracy scores for the BERT model not pretrained is only 82.73%.

4.4 Additional results not present in the original paper

We present two novel extensions to improve the classification accuracy on the dataset. First, we explore pretraining on a different dataset. Second, we experiment with a different model other than BERT. In both cases, we see these extensions can further improve inference and provide better understanding of indirect answers.

New Dataset. We incorporate The Stanford Natural Language Inference Corpus (Bowman et al. 2015) as an additional dataset to pretrain the BERT model on. Like MNLI, the SNLI dataset is used to train a model for the textual entailment task. We picked SNLI as our additional dataset to train on because we wanted our additional corpus to also focus on entailment, and because SNLI provided an alternate, more restricted corpus that could better prove the usefulness of pretraining, even if the dataset is smaller. We predicted that the SNLI-pretrained dataset would perform worse than the other pretrained models, but will still perform better than the BERT model trained on Circa only. From our experiments, we see that the SNLI-pretrained model does comparatively similar or better than the models pretrained on other datasets (MNLI, DIS, BOOLQ) in the different settings (strict/relaxed and unmatched/matched). For example, in the relaxed, matched setting, we saw that the SNLI-pretrained BERT model beat out the MNLI-pretrained BERT model by nearly 2%. Similar gains were made in the other experimental settings further shows that pretrained models provide an advantage in the entailment classification task, even when the dataset may be restricted. Furthermore, this shows that similar entailment tasks contain transferable knowledge, even if the tasks are not directly related to one or another.

New Architecture. Our second experiment consisted of using the BART (Lewis et al. 2019) for sequence classification. We chose to use BART because of its encoder-decoder architecture, in addition to the fact that the model was previously trained on MNLI and similar datasets. For this experiment, our hypothesis was that we expected BART would perform better than BERT on the Circa dataset. We trained BART using only the Circa dataset for the strict and relaxed matched case only, where we found that BART outperformed the other models, even those pretrained on other related datasets (MNLI, BOOLQ, DIS). For example, when looking at the matched relaxed setting, the BART model had a test accuracy of 85.41%, which beats the other models such as BERT-DIS-YN with a test accuracy of 83.89%. This shows that models with encoder-decoder architectures may be better suited for sequence classification and that more work needs to be done in this area to verify how much of an improvement these types of models might show.

5 Discussion

The larger implications of the experimental results show that training models to understand indirect answers is possible. For instance, even the strict setting with 9 labels ended with the lowest test accuracy of 74% for the models trained on Circa, which showed that the model was able to learn the representations of the Circa dataset quite well. Even the relaxed setting with 6 labels had a high accuracy, with the accuracy being 83.89%. However, the original paper by itself was not reproducible completely. Since 2 of our 3 claims were disproven (even with a large margin for error), there are parts of the experiment that needed to be fleshed out more in order to reproduce the training. The factors that made it irreproducible was the number of experiments to keep track of, the lack of clarity on which experiment was left out on each of the 10 experiments in the Circa unmatched case, and the training time that was necessary for the models. For instance, the number of experiments made it difficult to figure out what was needed for each step. And finally, the time stated in the paper was simply 30 minutes for training on a Cloud TPU. Since this statement was vague, we assumed that all experiments would take 30 minutes. However, we found this incredibly misleading since many of our experiments ended up taking far longer than 30 minutes, even with other GPUs.

5.1 What was easy

The authors of this paper provided sufficient explanations on what models were trained and which datasets were used, which made many portions of the paper easily reproducible. Specifically, we benefited from the authors' use of a commonly used model in BERT, which allowed us to use the HuggingFace library on BertForSequenceClassification. The availability of publicly packaged models helped ensure we did not have any reproducibility errors in the model itself, even when implementing the entirety of the testing from scratch. Furthermore, the author relied on publicly available datasets to conduct their experiments. Since the datasets were publicly available, we had a relatively simple time obtaining this information and cleaning the data for training and validating.

5.2 What was difficult

The difficulty of reproducing this paper stemmed from the number of experiments conducted. Overall, there were 8 different models we had to train: MNLI, BOOLQ, BERT-YN (question), BERT-YN (answer), BERT-YN (question answer), BERT-MNLI-YN, BERT-DIS-YN, and BERT-BOOLQ-YN. MLNI and BOOLQ did not need to be trained on Circa, but the remaining 6 models had to. For each model trained on Circa, we had two experimental settings: matched and unmatched. The matched setting only required one experiment (since there was only one split on the dataset: 60-20-20 training-dev-test), but for the unmatched setting, 10 different experiments were needed. Therefore, 6 + (6 * 10) = 66 experiments were conducted for one definition of labels. Since we had two different label definitions, strict and relaxed, we had to train a 66 * 2 = 132, plus an extra 2 from MNLI and BOOLQ models, giving us a total of 134. Training and testing all of these models took a long time, although it was feasible across a few days and split among multiple GPUs.

5.3 Recommendations for reproducibility

Our recommendations for the authors to improve the reproducibility are as follows:

- Adding original code to reproduce these results would be a tremendous asset to evaluate and ensure that the results are correct. While many of the methods employed to train these models are rather simple, since there are so many models and experiments to reproduce, having to create these from scratch makes it far more difficult.
- Creating more transparent numbers for how long each experiment ran to train, validate, and test would make it more transparent on how long an experiment might take and thus the overall train time.
- Describing which scenario was left out for each of the 10 experiments in the unmatched case for the Circa dataset. We assumed that for each experiment, we left a different scenario out (since there coincidentally were 10 scenarios), but we weren't sure if that was actually the case.

Communication with original authors

After reading the paper, we were concerned about being able to acquire the rewritten version of the Circa dataset, since the paper mentioned they converted questions into statements in order to help with accuracy. We reached out to one of the authors, Annie Louis, and she told us that the rewritten Circa dataset was not used for training, but only for testing the baseline out of the box MNLI classifier.

References

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. "The SNLI Corpus." (2015).

Clark, Christopher, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. "BoolQ: Exploring the surprising difficulty of natural yes/no questions." arXiv preprint arXiv:1905.10044 (2019).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).

Nie, Allen, Erin Bennett, and Noah Goodman. "DisSent: Learning sentence representations from explicit discourse relations." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4497-4510. 2019.

Williams, Adina, Nikita Nangia, and Samuel R. Bowman. "A broad-coverage challenge corpus for sentence understanding through inference." arXiv preprint arXiv:1704.05426 (2017).

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac et al. "HuggingFace's Transformers: State-of-the-art natural language processing." arXiv preprint arXiv:1910.03771 (2019).