# Replicating the Circa paper's experiments; Training the BERT model!

**MohammadHossein Rezaei**
University of Arizona
mhrezaei@arizona.edu

## Abstract

This document is a report on replicating the experiments done in the (Louis et al., 2020) paper to train models for understanding indirect answers, where the authors collected and annotated 34,268 pairs of questions and answers into a data set named "Circa." In the replications, the same models are trained and the accuracy is evaluated, which turns out to be very similar to the results of Louis et al., 2020.

## 1 Introduction

In order to replicate the experiments, the Circa data set is loaded from HuggingFace[1] and then preprocessed to be used for training BERT using the Transformers[2] library. After training the model, we use the trained model to predict the labels of the test data set to determine the accuracy. Steps to do the replications are relatively followed by Transformers' task guides for text classification [3], using the PyTorch[4] framework. All the codes used for training and testing can be found in the replicate Circa repositories[5] on my Github(username: brainrez). By the way, models are available on my HuggingFace models [6] and will be pointed out in the associated section. Same as Louis et al. (2020), replicating the experiments is divided into *matched* and *unmatched* setup and divide the experiment into *relaxed* and *strict* labels.

## 2 Preprocess the Circa

"In the matched setup, we assume that the response scenario is seen during training (randomly dividing our corpus examples into 60% training, 20% each for development/test)."(Louis et al., 2020) dataset_loader() function loads the Circa data set and splits it using the train_test_split method by setting the seed value equal to 42. Note that the examples without a majority label or marked as "other" are ignored. The experiments data sizes are:

| Experimental Settings | Train | Dev. | Test |
|---|---|---|---|
| RELAXED-matched | 19795 | 6599 | 6599 |

Then, the input is tokenized with the following tokenizer

```
AutoTokenizer.from_pretrained(
␣␣␣␣"bert-base-uncased"
␣␣␣␣).
```

Note that to train the model with both questions and answers, we need to tokenize both of them with a separator between them, which is achieved by providing both strings as the arguments.

## 3 Train the model

BERT model is trained using the Transofmers'

```
transformers.AutoModelForSequenceClassification(
␣␣␣␣'bert-baseuncased'
␣␣␣␣)
```

The following hyperparameters are used for fine tuning with *relaxed*, which are relatively[7] the same as the Circa paper.

| Model | Learning rate |
|---|---|
| BERT-YN (Question only) | 2e-5 |
| BERT-YN (Answer only) | 2e-5 |
| BERT-YN | 3e-5 |

For all the models, number of epochs are 3 and batch sizes are set to 32.

[1] https://huggingface.co/

[2] https://huggingface.co/docs/transformers/index

[3] https://huggingface.co/docs/transformers/tasks/sequence_classification

[4] https://pytorch.org/

[5] https://github.com/brianrez/replicateCirca

[6] https://huggingface.co/mhr2004

[7] For BERT-YN (Question only), the learning rate used is 2e-5, but Louis et al., 2020 used 3e-5.

# 4 Results for Relaxed Labels

## 4.1 Matched

In the following sections, we will briefly describe the model trained and compare the results with the Circa paper for the matched settings. Results are compared with Table 9 of Louis et al. (2020).

### 4.1.1 Majority class

For this baseline, no pretrained model was used[8] but a simple function was used to check if the restricted label is a "Yes" or not. [9] Note that Louis et al. (2020) argue that the majority of answers were "Yes". The accuracy results are as following:

| Model Trainer | Dev./valid. | Test |
|---|---|---|
| Louis et al. (2020) | 50.2 | 49.3 |
| M.H. Rezaei | 50.77 | 50.74 |

### 4.1.2 BERT-YN (Question only)

For this model, BERT is finetuned only on Circa's questions. [10] For testing the model on Test data set, Transfromers' pipeline is used to classify the input and determine the accuracy. [11] The accuracy results are as following:

| Model Trainer | Dev./valid. | Test |
|---|---|---|
| Louis et al. (2020) | 56.4 | 56.0 |
| M.H. Rezaei | 57.54 | 57.77 |

### 4.1.3 BERT-YN (Answers only)

Similar to the question only, BERT is finetuned only on Circa's answers. [12] For testing the model on Test data set, Transfromers' pipeline is used to classify the input and determine the accuracy. [13] The accuracy results are as following:

| Model Trainer | Dev./valid. | Test |
|---|---|---|
| Louis et al. (2020) | 83.0 | 81.7 |
| M.H. Rezaei | 82.21 | 81.80 |

### 4.1.4 BERT-YN (Question + Answer)

For this model, BERT is finetuned on both questions and answers. As mentioned in Section 2, inputs are tokenized sperately, which puts a [SEP] between the question and the answer. [14] For testing the model on Test data set, there are two approaches. First, using the Transfromers' pipeline to classify the input and determine the accuracy. [15] However, in this method, pipeline only accepts one string input and we have to manually put [SEP] between the question and answer in our input. In the second approach, we manually tokenize the question and answer (just like the training preprocess), then pass inputs to the model and return the logits to get the class with highest probability. [16] Inference and evaluate the accuracy on this approach takes a much longer time to process. The accuracy results are as following:

| Model Trainer | Dev./valid. | Test |
|---|---|---|
| Louis et al. (2020) | 88.4 | 87.8 |
| M.H. Rezaei(Approach 1) | 87.65 | 84.29 |
| M.H. Rezaei(Approach 2) | 87.65 | 87.12 |

# References

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7411–7425, Online. Association for Computational Linguistics.

[8] For more convenience, the dummy classifier on sklearn can be used.

[9] The code used for this experiments is available on Github repositories, titled "test_Majority.py".

[10] The code used for this experiments is available on Github repositories, titled "BERT_YN_Qonly.py". Also, the model is available on HuggingFace titled "mhr2004/BERT_Question_only".

[11] Code is titled "test_Qonly.py" on Github.

[12] The code used for this experiments is available on Github repositories, titled "BERT_YN_Aonly.py". Also, the model is available on HuggingFace titled "mhr2004/BERT_Answer_only".

[13] Code is titled "test_Aonly.py" on Github.

[14] The code used for this experiments is available on Github repositories, titled "BERT_YN_QA.py". Also, the model is available on HuggingFace titled "mhr2004/BERT_QandA".

[15] Code is titled "test_QandA_1.py" on Github.

[16] Code is titled "test_QandA_2.py" on Github.