Understanding Indirect Answers with BERT: A Study on the Circa Dataset

Udit Chaudhary

MS Data Science

Abstract

This study revisits the challenge of interpreting indirect responses to yes/no questions in dialogues, as explored in the paper “I’d rather just go to bed”: Understanding Indirect Answers. Using the Circa dataset, which contains 34,268 pairs of polar questions and indirect answers, we fine-tuned BERT models to classify these question-answer pairs into meaning categories. Our models achieved significant results, although there were notable differences compared to the original paper’s reported performance. This document details our approach, including data preparation, model setup, training, evaluation, and a comparison of our results with the reported benchmarks.

1 Introduction

The paper titled “I’d rather just go to bed”: Understanding Indirect Answers addresses the challenge of interpreting indirect responses to yes/no questions in dialogues. The authors introduce the Circa dataset, a large-scale English corpus containing various types of indirect answers. This study aims to replicate and extend the findings by implementing BERT-based models for sequence classification on the Circa dataset, MultiNLI and Boolq focusing on matched settings.

2 The Circa Corpus

Circa (meaning approximately) is our crowdsourced corpus designed for research on indirect answers. It consists of 34,268 question-answer pairs, featuring 3,431 unique questions with up to 10 answers each. Each pair is annotated with meaning based on a fixed set of categories.

The dataset comprises very short dialogues, each containing a question and its indirect answer. The texts vary in both semantic and syntactic forms and are grounded in 10 different situational contexts.

3 Data Preparation

2.1 Dataset

The Circa dataset comprises 34,268 pairs of polar questions and indirect answers, annotated with various meaning categories such as yes/no, uncertain, middle-ground, and conditional answers. The dataset was created via crowd sourcing to capture natural language usage.

2.2 Gold standard labels

Each (question, indirect answer) pair was evaluated by five annotators. We use the majority judgement as the gold standard, provided that at least three annotators agreed.

We employ an aggregation scheme. A RELAXED label is computed by collapsing the uncertain classes with the definite ones: ‘probably yes / sometimes yes’ → ‘yes’, ‘probably no’ → ‘no’, and ‘I am not sure how X will interpret Y’s answer’ → ‘In the middle, neither yes nor no’. We now analyze the distribution of the gold standard labels in Table 1.

Label	*Count*	*Percentage*
Yes	16628	50.40
No	12833	38.90
Yes, subject to some conditions	2583	7.83
In the middle, neither yes nor no	949	2.89

Table 1. Distribution of RELAXED gold standard labels.

2.3 Splitting

The dataset was split into 60% training, 20% development, and 20% test sets. This division ensured a robust evaluation framework while maintaining sufficient data for training and validation.

3 Models

We used the BertForSequenceClassification model from the Transformers library by Hugging Face. The model was configured for 4-class classification (yes, no, conditional yes, and in the middle). The pre-trained BERT model was chosen for its proven effectiveness in various NLP tasks and its ability to handle both contextual and sequential information within the text.

3.1 Related baselines and corpora

BOOLQ

BOOLQ is a question-answering dataset that focuses on factual yes/no questions (Clark et al., 2019). It pairs yes/no questions from web search queries with Wikipedia paragraphs that provide answers. The dataset contains 9.4k training examples, 3.2k development examples, and 3.2k test examples, with two target classes: 'yes' and 'no'. Achieving an accuracy of 66.3% without finetuning.

MNLI

The MultiNLI corpus (Williams et al., 2018) is a comprehensive dataset for textual entailment. It comprises premise-hypothesis sentence pairs, categorized into three target classes: 'entailment', 'contradiction', and 'neutral'. The dataset includes 392K training examples, 9K development examples, and 9K test examples. Achieving an accuracy of 27.8% without finetuning.

BERT-YN

BERT-YN is BERT fine-tuned solely on our Circa corpus (YN).Additionally, we explore the transfer of parameters learned from three related inference tasks. The model achieves a test accuracy of 89.8%.

BERT-BOOLQ-YN

Fine-tunes a BOOLQ model checkpoint (mhr2004/BERT_BOOLQ_Circa_YN) on our corpus with a new output layer. As BOOLQ is a Yes/No question answering system, even though it was developed for a different domain, we expect it to effectively learn the semantics of yes/no answers from this data. This model is a fine-tuned version of 'mhr2004/BERT_BOOLQ_Circa_YN' on the circa dataset, achieving a test accuracy of 94.5%. This model exclusively predicts the two classes 'yes' and 'no'.

BERT-MNLI-YN

BERT-MNLI-YN is first fine-tuned on the MNLI corpus, and then further fine-tuned on our YN data. This configuration examines whether the signals we aimed to capture with the out-of-the-box MNLI model (‘Abby-OGV/circa_mnli_yn’) can be enhanced by training on our target task. Our models for the MNLI task start from a BERT checkpoint from ‘Abby-OGV/circa_mnli_yn’ and reach a development accuracy of 89.5%. The model was trained using a learning rate of 5e-5, with a batch size of 16 over 3 epochs, fine-tuned specifically on the MNLI dataset before further fine-tuning on the Circa data. Our models for the MNLI task start from a BERT checkpoint from ‘Abby-OGV/circa_mnli_yn’ and reach a test accuracy of 89.5%. The model was trained using a learning rate of 5e-5, with a batch size of 16 over 3 epochs, fine-tuned specifically on the MNLI dataset before further fine-tuning on the Circa data.

3.2 Training Configuration

The following hyperparameters were used for training:

- **Learning rate:** 2e-5, which balances the trade-off between convergence speed and stability.
- **Number of epochs:** 3, sufficient to fine-tune the pre-trained weights without overfitting.
- **Batch size:** 16, to ensure efficient use of GPU memory while maintaining sufficient gradient updates per epoch.

4 Training Procedure

4.1 Fine-tuning

The BERT model was fine-tuned on the training set using the AdamW optimizer and a linear scheduler. Mixed precision training was employed to optimize performance and reduce memory usage. This approach leverages both 16-bit and 32-bit floating-point computations to speed up training and decrease memory footprint, enabling larger batch sizes and more efficient training.

4.2 Evaluation

The model was evaluated on the test set using classification metrics such as accuracy, precision, recall, and F1 scores. The classification_report from sklearn.metrics was used to calculate these metrics. This comprehensive evaluation provides insight into the model's performance across different classes, highlighting areas of strength and potential improvement.

	Model		
Epoch	**BERT-YN (Question only)**	**BERT-YN (Answer only)**	**BERT-YN**
		Average Training Loss	
1/3	0.5762	0.9755	0.5405
2/3	0.4076	0.9059	0.2970
3/3	0.2908	0.8550	0.1723

Table 2. Average Training Loss for the models at different epoch values

5 Experiment Results

5.1 Matched Setting

In the matched setting, one model was trained to get the numbers of one row. The model was trained and tested using 60% of the Circa corpus as training data, 20% as development data, and the remaining 20% as test data. This setup ensures that the model has seen similar scenarios during training and testing, providing a reliable measure of its performance in familiar contexts.

5.2 Unmatched Setting

In these experiments, the unmatched setting was not considered. This decision was made to focus on the model's performance within scenarios it has been trained on, ensuring an accurate comparison with the original paper's reported results.

6 Results

6.1 Matched Setting:

Model	Matched Setting				
	Accuracy	Test F-Score			
	Test	Yes	No	C.Yes	Mid
Baselines (no finetuning)					
MNLI	27.8	39.0	24.65	0	4.29
BOOLQ	66.3	71.3	59.2	0	0
BERT finetuned on Question or on Answer					
BERT-YN (Question only)	55.9	64.7	50.7	2.5	5.5
BERT-YN (Answer only)	81.7	84.0	80.4	89.0	23.5
BERT finetuned on Question + Answer					
BERT-YN	87.5	89.5	87.8	88.5	33.0
BERT-MNLI-YN	89.5	91.1	90.1	93.1	35.0
BERT-BOOLQ-YN	94.5	95.7	95.2	94.1	58.3

Table 9: Performance on the relaxed labels

7 Comparison with Original Paper

Our results show a similar trend to the original paper, with the BERT models achieving high accuracy, especially when using both the question and answer for classification. However, there were some differences:

7.1 Result Summary:

- **Baselines (no finetuning)**
  - MNLI: The accuracy decreased from 28.9% (Original) to 27.8% (Replication).
  - BOOLQ: The accuracy increased from 62.7% (Original) to 66.3% (Replication).

- **BERT finetuned on Question or on Answer**

    - BERT-YN (Question only): The accuracy slightly decreased from 56.0% (Original) to 55.9% (Replication).
    - BERT-YN (Answer only): The accuracy remained the same at 81.7% in both the Original and Replication studies.

- **BERT finetuned on Question + Answer**
    - BERT-YN: The accuracy slightly decreased from 87.8% (Original) to 87.5% (Replication).
    - BERT-MNLI-YN: The accuracy increased from 88.2% (Original) to 89.5% (Replication).
    - BERT-BOOLQ-YN: The accuracy significantly increased from 87.7% (Original) to 94.5% (Replication).

7.1 Accuracy:

Our models achieved better accuracy in terms of BERT-MNLI-YN which is first fine-tuned on the MNLI corpus and BERT-BOOLQ-YN which is first fine-tuned on the BOOLQ corpus, both followed by our YN data, by 1.3% and 7.4% compared to the original paper. This could be due to differences in model configuration in the checkpoints obtained from the Hugging Face pretrained models.

7.2 Class Imbalance:

Handling the class imbalance in the dataset posed challenges, which might have affected the performance on certain classes.

7.3 Model Configuration: Differences in hyperparameter tuning and training strategies could also contribute to the variations in results.

8 Discussion

Replication of Results

To replicate the highlighted results from the original paper, we followed these steps:

1. Data Preparation:

    - Downloaded the Circa dataset from Hugging Face Datasets.
    - Split the data into training, validation, and test sets.

2. Model Setup:
    - Used the BertForSequenceClassification model from the Transformers library by Hugging Face.
    - Configured the model for 4-class classification.

3. Training:
    - Fine-tuned the model on the training set using the specified hyperparameters.

4. Evaluation:
    - Evaluated the model on the test set.
    - Used classification_report from sklearn.metrics to calculate accuracy, precision, recall, and F1 scores.

9 Difficulties Encountered

Replicating the code posed challenges mainly in preprocessing. Adapting the data for BERT required meticulous tokenization and encoding, ensuring alignment with the model's input requirements. Handling label filtering and ensuring compatibility with BERT's format posed initial hurdles, demanding careful attention to detail and alignment with the model's specifications.

10 Conclusion

Our study demonstrates that BERT models are effective for classifying indirect answers to yes/no questions in dialogues, achieving high accuracy on the Circa dataset. However, there are notable differences in performance compared to the original paper, likely due to variations in preprocessing, model configuration, and handling of class imbalance. Future work could explore advanced preprocessing techniques and further hyperparameter tuning to bridge this performance gap.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186