Final Project Report

1 Introduction

Logical reasoning is involved in a wide range of NLP tasks, but it still remains unclear about whether they are actually reasoning. Therefore, a research question naturally arises: can neural networks be trained to conduct logical reasoning presented in natural language?

The paper '*On the Paradox of Learning to Reason from Data*'[1] tried to answer this question. In this paper, they defined a confined problem space containing simple questions of propositional logic, and sampled datasets from the problem space using two different sampling strategies RP(Rule Priority) and LP(Label Priority). By training a BERT base model on these two datasets, they found out that the model trained on RP train set performs well on the RP test set, but performs poorly on the LP test set. On the other hand, the model trained on LP train set performs well on the LP test set, but performs poorly on the RP test set. This phenomena is peculiar, in the sense that both RP and LP belongs to the same problem space, and if the model generalizes well on one, it should do well on the other, which is contrary to the actual observations. Then, the paper moved on to prove that the model has not learned to emulate the correct reasoning function, and that the model is learning the statistical features.

The results of the paper is quite interesting, therefore I wanted to make some further explorations based on this paper. In this report, I will focus on two questions that I came up with after reading this paper:

1. Is this problem particular to BERT base model? Is this phenomena an architecture issue?
2. Is it the model's inability to 'learn' how to reason, or the dataset's inability to 'teach' model how to reason?

2 Model Architecture

Because the paper only used BERT (encoder-only) throughout the experiments, I was suspicious about whether this issue exists in other model architectures. Therefore, I tried to reproduce the results on other architectures, specifically GPT2 (decoder-only) and T5 (encoder-decoder).

Due to limited computation resource and limited time, I wasn't able to perform the reproduction at full scale. According the original paper, training a BERT base model on RP/LP takes less than 2 days on 4 NVIDIA 1080Ti/2080Ti GPUs[1]. Therefore, I cut down the size of RP and LP (280k examples each) to 28k examples each. Other training configurations are mostly identical to the original paper, which is 20 training epochs, a learning rate of 4e-5, warm-up ratio of 0.05, and a batch size of 16 (different from the original paper due to not enough GPU memory). Unless mentioned, all models trained in this report are trained under identical configurations.

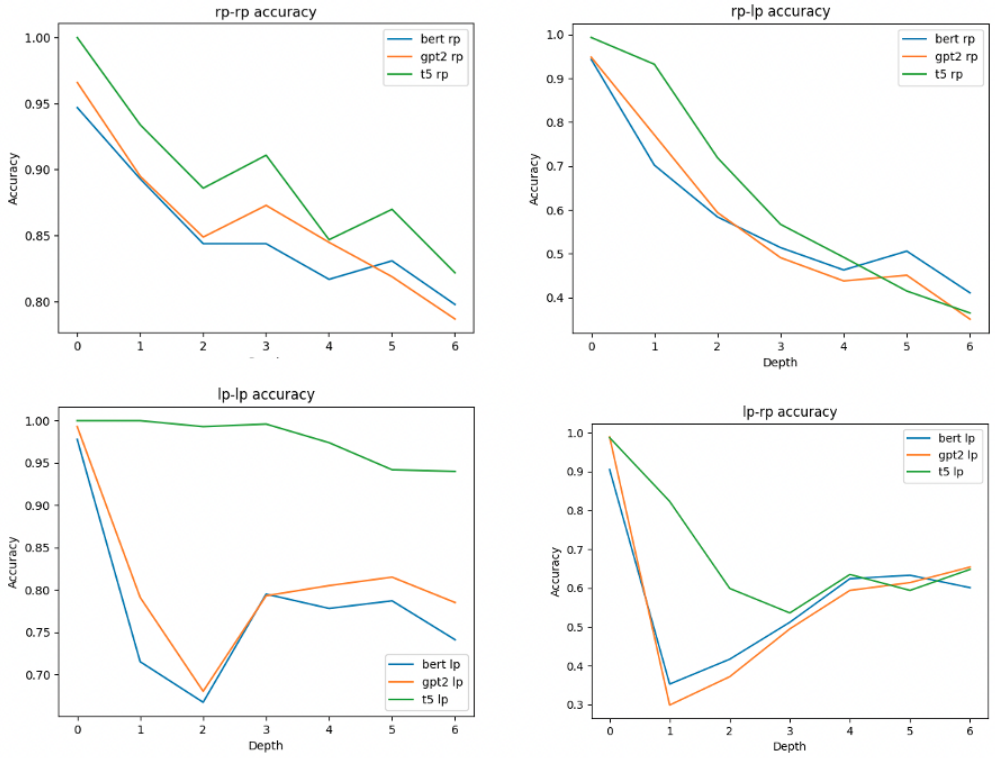

Fig.1 Test accuracy on LP/RP for the BERT/T5/GPT2 model trained on LP/RP. The dataset before the dash is the dataset the model is trained on, and the dataset after the dash is the dataset the model is tested on, e.g. the 'rp-lp accuracy' chart shows the results of models trained on RP train set and tested on LP test set. Note that the RP/LP dataset I used is a scaled-down version of the original RP/LP dataset. For each architecture, the model with the least parameters (i.e. base model) is used.

As we can see from Fig.1, even though I cut down the dataset size, the performance of the BERT base model trained on RP attains high accuracy on RP test set, but the accuracy drops significantly on the LP test set (and vice versa), similar to what is observed in the Table 1 of the original paper[1]. The issue also happens on the GPT2 model and the T5 model, showing that this phenomena is not particular to the BERT base model, and instead it is a common problem that different architectures share.

Train	Test	0	1	2	3	4	5	6
RP	RP	99.9	99.8	99.7	99.3	98.3	97.5	95.5
	LP	99.8	99.8	99.3	96.0	90.4	75.0	57.3
LP	RP	97.3	66.9	53.0	54.2	59.5	65.6	69.2
	LP	100.0	100.0	99.9	99.9	99.7	99.7	99.0

Table 1: Test accuracy on LP/RP for the BERT model trained on LP/RP; the accuracy is shown for test examples with reasoning depth from 0 to 6. BERT trained on RP achieves almost perfect accuracy on its test set; however the accuracy drops significantly when it's tested on LP (vice versa).

Table 1 of the original paper[1]

3 Dataset Structure

The second question originated after I read a similar paper, '*Transformers as Soft Reasoners over Language*'[2]. In this paper, the problem space is rather similar, consisting of simple logical reasoning problems.

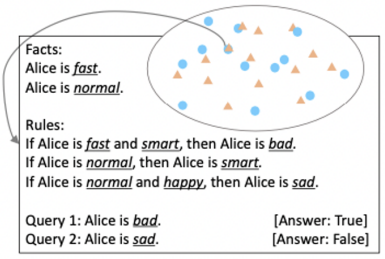

Figure 1: Problem setting: a confined problem space consisting of logical reasoning problems; dots and triangles denote examples sampled from different distributions over the same problem space.

(Input Facts:) Alan is blue. Alan is rough. Alan is young.
Bob is big. Bob is round.
Charlie is big. Charlie is blue. Charlie is green.
Dave is green. Dave is rough.
(Input Rules:) Big people are rough.
If someone is young and round then they are kind.
If someone is round and big then they are blue.
All rough people are green.

Q1: Bob is green. True/false? **[Answer: T]**
Q2: Bob is kind. True/false? **[F]**
Q3: Dave is blue. True/false? **[F]**

Figure 1: Questions in our datasets involve reasoning with rules. The inputs to the model are the context (facts + rules) and a question. The output is the T/F answer to the question. Here the underlying reasoning for the true fact (Q1) is: Bob is big, therefore rough (rule1) therefore green (rule4). Note that the facts + rules themselves change for different questions in the datasets.

Left Figure: Problem setting of [1]; Right Figure: Problem setting of [2].

In [2], a RoBERTa-large model is trained on a dataset that consists of questions sampled from the problem space (similar to [1]), and performs well on the dataset's test set, which is not a surprise. However, the model also generalizes well on out-of-distribution tests, and performs well even on hand-authored problems, which contradicts the results of [1]. I was confused about the contradiction, and compared the differences between [1] and [2].

After comparing these two papers in detail, I found that there is a huge difference in how many rules can be in a single example. In [2], each example contains 1–9 rules, while in [1], each example contains 0–120 rules, which is a stark contrast.

Therefore I suspected that it was the dataset's fault, instead of the model's inability that caused the model to fail in emulating the correct reasoning function. The rules in each example was simply too much for a model to capture the underlying logic, causing the model to learn other irrelevant statistical features. It is like giving calculus textbook to first-graders in primary school: all they can do is making guesses using the patterns of the perplexing notations, but they couldn't really learn the math behind it, and apparently you couldn't say that it is the student's fault. I then proceeded to conduct experiments to support my viewpoint.

3.1 Experiments

3.1.1 Experiment 1

The first experiment involves separating the examples based on the number of rules and examining the model's accuracy on examples with different rule counts. In this experiment, I used the BERT base model trained on LP, and took 20 rules per example as a threshold.

Figure 2: Accuracy of BERT base model trained on LP, on RP/LP dataset, separated by number of rules per example. The blue line in the first figure shows the accuracy of the model on the examples that holds no more than 20 rules in the RP test set, the blue line in the second figure shows the accuracy of the model on the examples that holds no more than 20 rules in the LP test set. The orange line in the first figure shows the accuracy of the model on the examples that holds more than 20 rules in the RP test set, the orange line in the second figure shows the accuracy of the model on the examples that holds more than 20 rules in the LP test set. The blue line in the third figure is the difference between the blue line in the second and the first figure, and the orange line in the third figure is the difference between the orange line in the second and the first figure.

It is observed that there is a more significant accuracy drop for examples with more than 20 rules than examples with no more than 20 rules, and on the LP test set, the accuracy on examples with more than 20 rules and reasoning depth larger than 4 somehow surpasses the accuracy on examples with no more than 20 rules. These observations suggest that the model overfits the examples with more rules to a certain degree.

3.1.2 Experiment 2

In the second experiment, I constructed a simple dataset which is a subset of the original problem space, containing extremely simple rules that forms a logic chain in which the given fact would be at the starting end of the logic chain and the query at the end of the chain.

A more formalized description would be: for a example with reasoning depth h, each example consists of $h+1$ atoms ($\{a_1, ..., a_{h+1}\}$), h rules ($\{a_1 \to a_2, ..., a_h \to a_{h+1}\}$), 1 fact (a_1) and 1 query (a_{h+1}). Apparently, I can reason all the way through the chains of rules easily from a_1 to a_{h+1}.

In addition to this, I also constructed a dataset with m additional disturbance rules, and each example would consist of $h+2m+1$ atoms ($\{a_1, ..., a_{h+1}, b_1, b_2, ..., b_{2m-1}, b_{2m}\}$), $h+m$ rules ($\{a_1 \to a_2, ..., a_h \to a_{h+1}, b_1 \to b_2, ..., b_{2m-1} \to b_{2m}\}$), 1 fact (a_1) and 1 query (a_{h+1}). The original chain of rules is still preserved, the only difference is that some irrelevant rules are added to mislead the model's judgement.

Then, I tested the performance of the BERT base model trained on RP on this simple dataset I constructed, and the results are shown in Figure 3.

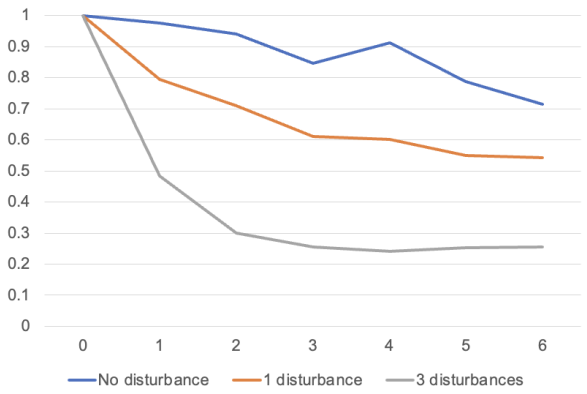

Figure 3: The model's (BERT base model trained on RP) accuracy on the constructed dataset with different numbers of disturbance rules.

Without the disturbance rules, the model did well, but with just 1 disturbance rule added, the accuracy drops by about 20%, and the drop is about 50% with 3 disturbance rules added. This shows that the model lacks robustness to those disturbance rules, which might be a result of learning the unrelated statistical features.

3.2 Improvement

With the experiments, it was clear that the model is terribly overfitted to irrelevant statistical features that it is not resilient to even the smallest disturbances. So, what can we do to improve the model's ability to emulate reasoning, using RP/LP?

The straightforward idea is to imitate [2] and train the model on examples that has lesser rules, making the dataset more 'learnable'. Therefore, I moved on to train the BERT base model on examples sampled from the problem space using RP, but examples with more than 30 rules are abandoned. The newly constructed dataset, which I would refer to as RP_LIM, has the same amount of examples as in RP.

In addition, because examples in RP_LIM has lesser rules, it is hard to get examples of deeper reasoning depth, and the positive and negative examples are seriously disproportionate. So when constructing RP_LIM, I made further restrictions to adjust the ratio of the examples of different reasoning depths and balance the proportion of the positive and negative examples.

After training a BERT base model on RP_LIM, I tested the model under the same conditions as in Fig.3.

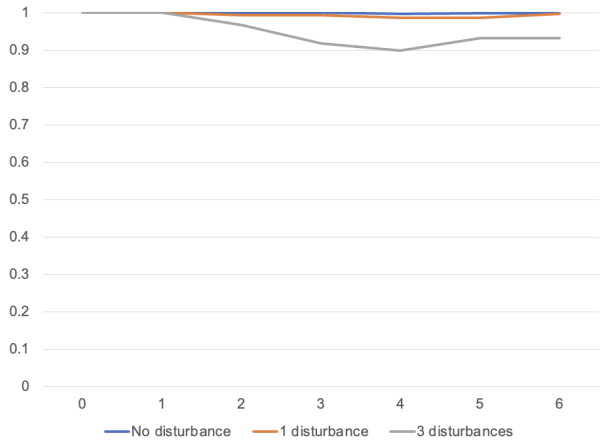

Figure 4: The accuracy of the RP_LIM trained BERT base model on the constructed dataset with different numbers of disturbance rules.

As we can see in Fig.4, comparing with Fig.3, the model trained on RP_LIM gets far higher accuracies and exhibits far better robustness over disturbances.

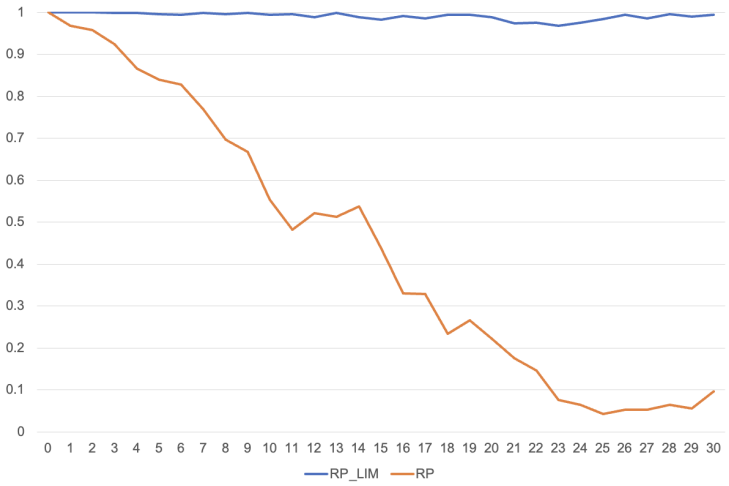

Figure 5: The accuracy of the RP_LIM/RP trained BERT base model on the constructed dataset with deeper reasoning depths. Note that no disturbance rules are added.

Furthermore, the model trained on RP_LIM performs well even to constructed examples with deeper reasoning depths, as shown in Fig.5, which is similar to the results in [2]. Therefore, it can be shown that the approach of limiting the complexity of the training examples significantly improves the model's ability to generalize, providing a strong evidence that the model's inability to emulate the correct reasoning function has a lot to do with the overly complex dataset.

4 Further Discussions

Although the main focus of paper [1] was on whether neural networks can be trained to conduct logical reasoning presented in natural language, there is another perspective to its findings.

In the paper, RP and LP are distinct sampling methods used to sample examples from the problem space. It occurred to me that the different approaches we use to collect data from various sources are, in essence, analogous to these methods. For example, when collecting images to train a diffusion model, the problem

space would encompass all possible images, the process of collecting them is essentially a sampling process, and collecting those pictures from Google or from Baidu can be seen as different ways of sampling.

As in the paper, the model trained on RP doesn't perform well on LP, and vice versa, and through my experiments, we can see that the complexity of the dataset plays a crucial role in this issue. Therefore, I believe the key takeaway is that the complexity of a dataset should be carefully considered when constructing it. It's important to assess how well-suited a dataset is for training a model, or else the model is likely to overfit irrelevant statistical features introduced by the sampling method.

To my knowledge, there has been limited research exploring this issue. As such, I believe it would be innovative to try to quantify both the complexity of a dataset and its ability to effectively train the models. This could lead to clearer principles that researchers can follow when designing datasets, ultimately improving the quality and generalizability of machine learning models.

References

[1] Zhang, H., Li, L. H., Meng, T., Chang, K.-W., & Van den Broeck, G. (2022). *On the paradox of learning to reason from data*. arXiv. https://arxiv.org/abs/2205.11502

[2]Clark, P., Tafjord, O., & Richardson, K. (2020). *Transformers as soft reasoners over language*. arXiv. https://arxiv.org/abs/2002.05867