# CS378: Final Project: Data Artifacts

# Improving the performance of the ELECTRA model using Dataset Classification

**Chin Kiat Tan**
The University of Texas at Austin, Computer Science
ct33726
ct33726@utexas.edu

Link to github repository: https://github.com/tanchinkiat99/nlp-data-artifacts

**Abstract**

With the increase in the complexity of models in natural language processing (NLP) comes a growing focus on the data used. In this paper, we explore the possibility of improving the efficiency of training by modifying the datasets used for training. By using *confidence* as a metric to evaluate the difficulty of the model learning how to predict each data point, we classify the data points into two classes – *easy to learn* and *hard to learn*. We can then form modified datasets which contain different proportions of each class of data points, and look for a trend which could point to certain combinations that increase the effectiveness of training. The results show that model accuracy could be improved by adjusting these proportions in the training data used.

## 1 Introduction

In this project, I will be analyzing the performance of the ELECTRA model (google/electra-small-discriminator), evaluating its shortcomings on NLI and using dataset classification to improve its performance.

In machine learning, and NLP specifically, models are typically trained on a particular dataset, and their performance is evaluated on a separate validation or test dataset. However, in many practical applications, the distribution of the data may shift over time, leading to out-of-distribution (OOD) samples that the model has not seen before (Linzen, 2020). This can result in a drop in the model's performance, as the model may make inaccurate predictions on these new data points.

Out-of-distribution generalization is important for real-world applications, as it enables models to make accurate predictions on new and unseen data. While the models' mechanisms are crucial in enabling this, we also need to look at the datasets used. Not all data examples may be as useful to the model during training (Vodrahalli, 2018), so we need a way to categorize data into categories varying in usefulness for model training.

Dataset cartography refers to the process of mapping and visualizing data in a way that allows individuals to gain insights into complex data sets (Swayamdipta et al., 2020). It involves the use of cartographic techniques and tools to create visual representations of data that can help users understand the spatial and temporal patterns, relationships, and trends within the data.

Dataset cartography has been used to map data into a coordinate space to construct data maps (Swayamdipta et al., 2020) using certain training dynamics. Specifically, the mean and standard deviation of the gold label probabilities can be used as a measure of the confidence and variability of the data points. The data points can then be categorized as easy to learn (high confidence and low variability), hard to learn (low confidence and low variability) and ambiguous (high variability).

This project will take a simplified approach: we will use a single metric, the confidence to classify the data into 2 categories – **easy to learn** (high confidence) and **hard to learn** (low confidence).

Intuitively, data points that are hard to learn would be more useful during training in helping the model generalize for downstream tasks where most of the data could be unseen in training data. We will train the ELECTRA model on datasets which contain different proportions of these 2 classes of data points, then evaluate the efficacy of training the ELECTRA model on these different subsets.

## 2 Methodology

Our goal is to first classify the dataset by mapping datapoints with respect to the confidence, then using either one or a combination of these subsets in training to improve the accuracy of the ELECTRA model. The task that we are training and testing the model on is Natural Language Inference (NLI).

We will split the data examples into two categories: easy to learn and hard to learn. Then, train the ELECTRA model on datasets that comprise different proportions of examples from these subsets, and evaluate the effect of this by comparing the accuracies of the model when trained on the different datasets.

### 2.1 Classifying the Dataset

To classify our dataset, we will make use of a single metric of the data points – confidence.

The confidence of a single example gives an indication of how confidently the model can correctly predict the label of the data point. A data point with high confidence suggests that the model learns how to predict it easily, and thus will be classified as easy to learn, while a data point with low confidence will conversely be classified as hard to learn.

The confidence is calculated by the mean probability of a correct prediction for that data point over every training epoch (Swayamdipta et al., 2020):

$$\hat{\mu}_i = \frac{1}{E}\sum_{e=1}^{E} p_{\theta^{(e)}}(y_i^*|x_i)$$

where $p_{\theta^{(e)}}$ is the model's probability of predicting the correct label $y_i^*$ for a given example $x_i$ with parameters $\theta^{(e)}$ at the end of the $e^{th}$ epoch (Swayamdipta et al., 2020).

### 2.2 Task: Natural Language Inference

Natural Language Inference (NLI) is an NLP task which involves determining the logical relationship between two sentences, a premise and a hypothesis. Based on a premise that provides contextual information, a hypothesis is a statement that must either be true, false or unknown.

The dataset that I will be using to evaluate the ELECTRA model on NLI is the Stanford Natural Language Inference (SNLI) Corpus (Bowman et al., 2015). This comprises 570,000 pairs of human-written English sentences, manually labelled with the 3 labels entailment, contradiction and neutral. Each pair of sentences have labels from 5 mechanical turk workers and a consensus label based on the majority label.

### 2.3 The ELECTRA Model

ELECTRA stands for Efficiently Learning an Encoder that Classifies Token Replacements Accurately (Clark et al., 2020), and is a model developed by a team of researchers at Google AI Language. This model is a language model for NLP tasks, particularly for text classification and generation. The model architecture is the same as BERT's and is based on the transformer architecture, but it differs many other NLP models in its pre-training objective. Instead of predicting the next word in a sequence (like in GPT-2) or Masked Language Modeling (MLM) used in the pre-training of BERT, ELECTRA uses a discriminator-generator approach to pre-training.

Instead of MLM, the pre-training task for ELECTRA is *replaced token detection*, where the model is trained to identify input tokens that have been synthetically generated. In this task, given an input sequence, a random set of positions are selected in which the tokens are replaced with a [MASK] token. A generator is then trained on predicting these original

masked-out tokens. This corrupted output is then fed as input into the discriminator which then learns to identify the tokens that have been replaced by the generator. Then, after pre-training, the generator is discarded and the discriminator is fine-tuned for downstream tasks.

This pre-training task has been proven to more sample-efficient than MLM, because it is defined over the entire input sequence rather than just the masked-out tokens. This targets an issue in BERT where the pre-training of the model is based on masked-out tokens, which are usually not present in downstream tasks and has shown to produce higher accuracy on downstream tasks than BERT.

| **Model Version** | **O** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Size of training dataset** | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| **Proportion of "easy-to-learn" examples** | 0.638 | 0.100 | 0.200 | 0.300 | 0.400 | 0.500 | 0.600 | 0.700 | 0.800 | 0.900 |
| **Proportion of "hard-to-learn" examples** | **0.351** | **0.900** | **0.800** | **0.700** | **0.600** | **0.500** | **0.400** | **0.300** | **0.200** | **0.100** |
| **Evaluation Accuracy (3 epochs)** | **0.665** | **0.489** | **0.532** | **0.545** | **0.520** | **0.583** | **0.653** | **0.700** | **0.620** | **0.510** |
| **Evaluation Accuracy (6 epochs)** | **0.753** | **0.512** | **0.583** | **0.562** | **0.558** | **0.630** | **0.762** | **0.727** | **0.633** | **0.567** |

*Figure 3: Performance of ELECTRA Model on modified training datasets, each with 10 epochs and with accuracy from evaluation on 300 test examples*

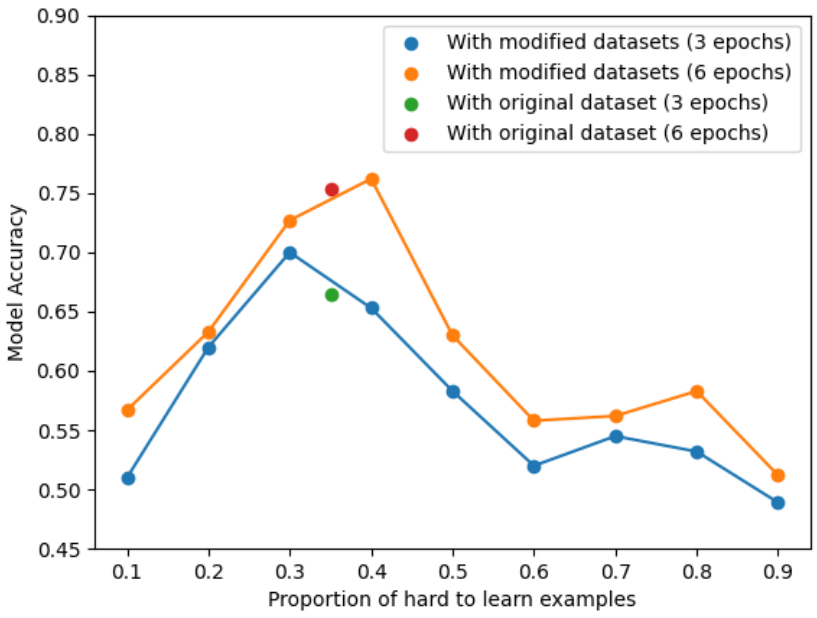


*Figure 4. Graph of Model Accuracy against proportion of "hard-to-learn" examples in training data*

## 3 Performance Analysis

### 3.1 Analyzing Confidence of Training Examples

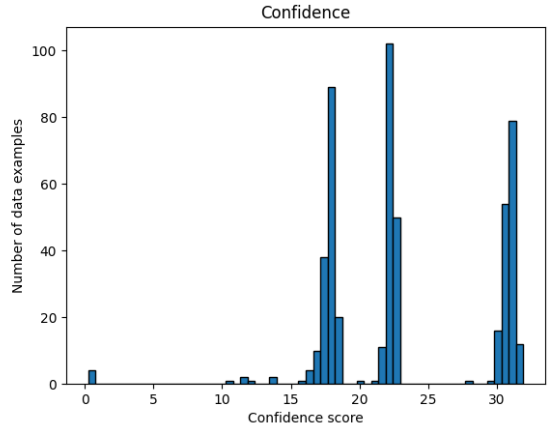


*Figure 1: Histogram showing distribution of confidence scores of data examples using a training dataset with 500 examples and 10 epochs*

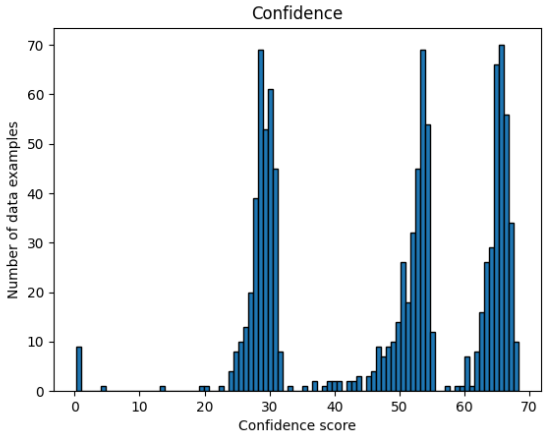


*Figure 2: Histogram showing distribution of confidence scores of data examples with a larger training set of 1000 examples and 20 epochs*

In figure 1, an initial calculation of the confidence levels was made using a small subset of the training data (500 examples) and the distribution of the confidence scores for the data examples was plotted on a histogram for observation of any trend that may be present. Note that the confidence scores (which are mean probabilities of correct label predictions over epochs) are mostly more than 1, as we are using the exponential of the mean probabilities to exaggerate and highlight any present trends in the distribution of the confidence values. We will refer to these exponentiated values as “confidence scores”.

There are 3 distinct peaks indicating 3 regions of confidence scores which are most common among the data examples, and this is also observed in Figure 2 where we do the same analysis on the entire training set.

We also observe that in Figure 2, the confidence scores are generally larger, which is likely due to a larger number of epochs used. The confidence scores of the region of the middle peak are also closer to that of the peak with the highest confidence scores as compared to Figure 1.

### 3.2 Analyzing Performance of ELECTRA Model on Modified Training Datasets

To get an initial gauge on performance, the ELECTRA model was trained on a subset of 1000 examples from the SNLI corpus and evaluated on 300 examples from the test set. The model achieved an average accuracy of around **70%.** This is without prior categorizing or any augmentation to the dataset.

The model was then trained on modified training datasets with varying proportions of each category of data examples.

Figure 3 shows the different variants of the models that were each trained with a different proportion of easy to learn and hard to learn examples. Note: *Model O* represents the original model that was trained on an unmodified dataset. Figure 4 presents the results in a graph and suggests a possible correlation between the proportion of hard to learn examples used in the dataset and the resulting model accuracy.

We can see that there might be a positive relationship between the proportion of hard to learn examples and the model accuracy up to a certain threshold (around 30% of hard to learn examples for 3 epochs of training, and 40% for 6 epochs) from which the model accuracy starts to diminish sharply when we continue to increase this proportion.

## 4 Improving the ELECTRA Model

### 4.1 Classification of Training Data

To categorize the dataset, we imposed a confidence score threshold where any examples above it will be classified easy to learn, since they have relatively higher confidence scores, and examples below it will be classified as hard to learn.

Based on the observed trends in the distribution of confidence scores, we will define a threshold such that the data examples in the 2 peaks with higher confidence scores will be classified as easy to learn, while those in the peak with lower scores will be classified as harder to learn. Additionally, to eliminate ambiguity in some examples that could be either easy to learn or hard to learn, we will extend this threshold to be a range of values, where the examples that lie within this range will not be considered as part of our new training set.

The final threshold used was a range of 40.0 to 45.0. More specifically, data examples with confidence scores below 40.0 (exclusive) are classified as hard to learn, while those above 45.0 are classified as easy to learn.

### 4.2 Applying Results to Improve Model Training

As seen in Figure 4, when both 3 and 6 epochs of training are used, we have areas which the lines showing the accuracy trend of the models trained on the modified datasets are higher than that of the green and red points which denote the accuracy of the model trained on the unmodified dataset with 3 and 6 epochs respectively. Thus, though the improvement is slight, this indicates a potential method to improve the model accuracy with dataset categorization.

## 5 Evaluating the Improvement

From the results, we managed to slightly improve the model performance by using the modified training datasets. The unmodified training data contained 35.1% hard to learn examples. When 3 epochs of training are used to train the model, we see that using a training dataset with 30.0% hard to learn examples (*Model G*) will improve the model accuracy by about 3.5%. If 6 training epochs are used, a training dataset with 40.0% hard to learn examples (*Model F*) will improve accuracy by about 0.9%.

This method of increasing the efficacy of training could have many potential benefits – less training resources required due to less training epochs required and better accuracy when working with limited data.

However, the improvements in accuracy are quite small and may require more in-depth testing to produce a viable improvement.

### 5.1 Limitations of Results

One limitation of this study is that the size of the dataset. Using datasets with only 1000 examples may limit the observation of trends in confidence scores and accuracy improvement. To increase the reliability of the results, a larger dataset should be used.

Another limitation is that we are deciding the threshold for confidence scores to classify the data arbitrarily. More research into choosing an appropriate confidence score threshold and calculating the confidence scores could help produce more significant improvements in model accuracy.

## 6 Conclusion

This project has given insight into the viability of improving the efficiency of training by classifying a dataset and using different proportions of data examples with varying “learning difficulties”. Though, there are limitations regarding the significance of the results and the methods used for the analysis. Nevertheless, it is interesting that the results somewhat prove our hypothesis and show potential of similar methods of improving efficiency of training.

In further research into the topic, one could dive into the possibility of using other metrics to evaluate the “learning difficulty” of data examples, in addition to confidence. An example is variability, which can be used as a measure of the variance of the probability of a correct prediction (Swayamdipta et al., 2020). More categories can also be introduced, instead of just a binary classification. Soft

reweighting of the data examples can also be explored instead of entirely changing the dataset composition, which could have eliminated some potential benefits of using all available data examples. The effect of such dataset classification could also be studied on other NLP tasks, such as Question Answering, Part-of-Speech (POS) tagging and Named Entity Recognition (NER).

## 7 Related Work

This project uses concepts and builds on the results of the paper, *Dataset cartography: Mapping and diagnosing datasets with training dynamics* (Swayamdipta et al., 2020).

It also heavily uses the starter code for using the ELECTRA-model provided by CS 378 (Bostrom et al., 2022).

## References


Kailas Vodrahalli, Ke Li, and Jitendra Malik. 2018. Are all training examples created equal? *An Empirical Study*. ArXiv:1811.12569.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *In Proceedings of the International Conference on Learning Representations (ICLR)*.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *ACL*.

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

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online, November. Association for Computational Linguistics.

Kaj Bostrom, Jifan Chen, and Greg Durrett. 2022. *Starter code for final project for CS 378 -- dataset artifact*, utcsnlp/cs378_fp, https://github.com/utcsnlp/cs378_fp