CleaNLP: Detecting Label Errors in General NLP Tasks

Vedang Lad
MIT
Cambridge, MA
vedlad@mit.edu

Alex Wang
MIT
Cambridge, MA
wang7776@mit.edu

Ryan Wilson
MIT
Cambridge, MA
ryanwils@mit.edu

Abstract

This paper presents a method for identifying label errors in natural language processing (NLP) datasets using the T5 model. The T5 model is a large-scale, multi-task language model that has been shown to achieve state-of-the-art performance on a variety of natural language understanding tasks. We used the T5 model to analyze a general dataset of labeled NLP examples and identified instances where the model predicted a different label than the one provided in the dataset. We found that the T5 model was able to accurately identify label errors in the dataset after finetuning on a T5 model, demonstrating the potential for using large-scale language models to improve the quality of NLP datasets.

1 Introduction

As data-centric artificial intelligence is becoming increasingly important in the modern world, there has been a greater demand for financial and energy optimization. While there are many ways to reinforce the performance of a model, such as through enlarging datasets, increasing computing power, algorithm tuning, cross-validation, etc., these can be very costly in terms of time, money, or energy.

Instead, we look to introduce an alternate form of optimization through detecting label errors in general NLP tasks. Oftentimes when there are mislabeled features that are detrimental to a given model, it can be worthwhile to focus on correcting these labels and improve the quality of the data being trained on rather than spending resources in other ways.

We seek to present a general error-finding algorithm that can be applied to a wide array of NLP tasks. We will utilize the pretrained knowledge in T5 to find errors in many types of NLP datasets, re-casting them as multitask classification tasks that T5 can handle, making use of the versatility of T5.

2 Motivations and Related Work

As first coined in the paper (Northcutt et al., 2019), using the probabilities to detect errors in machine learning datasets was generally called “Confident Learning". Using uncertainty calculations rooted in information theory, (Northcutt et al., 2019) showed how using just the predicted probabilities of a model and the target labels, one can discern a model’s confidence in the label. Using this confidence, one can determine whether a given label in a dataset has been mislabeled.

Figure 1 shows some example errors found in 2012 ILSVRC ImageNet. Using the idea presented in (Northcutt et al., 2019), we can see that the work described in (Wang and Mueller, 2022) uses the same uncertainty calculations, but applies them to NLP. In particular, the same uncertainty calculations are utilized for entity classification as seen by figure 2. While this proves useful for the problem of entity recognition, it does not address other NLP tasks. More specifically, there does not exist a general NLP label error detection method.

In order to “cast" an NLP task to a multi-class classification problem, we utilize the transformer architecture T5 (Raffel et al., 2019). Using transfer learning, this transformer architecture can handle a wide array of NLP tasks, taking an arbitrary NLP task (i.e summary, translate, etc) and producing a result. We take advantage of pretrained models in order to implement T5 within our algorithm. See the figure 3 to see the general capabilities of the transformer. Using T5, we can utilize the open-source package called cleanlab, which uses the same uncertainty calculations as in (Northcutt et al., 2019), to find errors in a data set.

Using the pieces described above, we create a general algorithm that in which a user inputs an NLP task and a corresponding dataset. Our algorithm utilizes the pretrained model of T5 to reframe the dataset as a classification task. We utilize

Figure 1: identified label issues in the 2012 ILSVRC ImageNet train set using CL as shown by (Northcutt et al., 2019)

existing open-source packages , in particular, the "find_label_issues" function from cleanlab, to find corresponding errors in the dataset. In order to boost error finding performance, we find that fine-tuning the dataset on T5 prior to error detection increases error finding performance.

3 Models and Methodology

For our model, we use a pre-trained T5 transformer as well as methods from cleanlab to find label issues. The T5 transformer is an encoder-decoder model pre-trained to handle general text-based NLP tasks by converting these tasks into a text-to-text format (Raffel et al., 2019). This conversion is done by adding the task type to the beginning of the input, and the resulting string can then be used as the input for T5.

Typically, language models are pre-trained on large unlabeled datasets such as Common Crawl. However, such datasets are often full of useless text like error messages, source code, duplicate text, etc. T5 is trained on the dataset C4, which is a cleansed version of Common Crawl that is still two orders of magnitude larger than Wikipedia. The authors of C4 took a 2019 scrape of Common Crawl and placed some filters on it removing any sentences without valid terminal punctuation, removing any duplicate text, source code, any pages with offensive language and several other restrictions.

We use this transformer as a sort of baseline solution for the labeled dataset that we desire to clean. By prepending the appropriate task to the inputs and finetuning T5 on the dataset in question, we are able to use the transformer to generate predicted

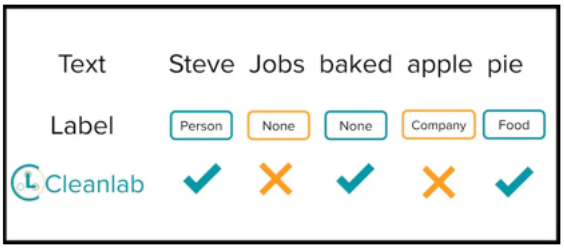

Figure 2: An example of an entity classification error detections as done by (Wang and Mueller, 2022)

labels for the inputs. We then pass the actual labels and the corresponding predictions into a function from cleanlab that produces a confidence score for each label, which we can use to narrow down the search for mislabeled data.

We evaluated the effectiveness of this approach by measuring the precision and recall for the detection of mislabeled data that we intentionally introduce to a dataset. We introduced mislabeled data by selecting a random sample of inputs that the model was already confident had been correctly labeled (having a label score of >0.99) and randomly swapped 30% their labels to be different. In this subset of data, we have knowledge of the true mislabels and can therefore compute how well our approach detects these errors. Our metrics for this were AUROC and AUPRC. We chose AUPRC for the case where we care more about having a very clean dataset and falsely predicting a correct label as an error is okay. In contrast, we chose AUROC for the case where we care equally between predicting correctly and incorrectly labeled data.

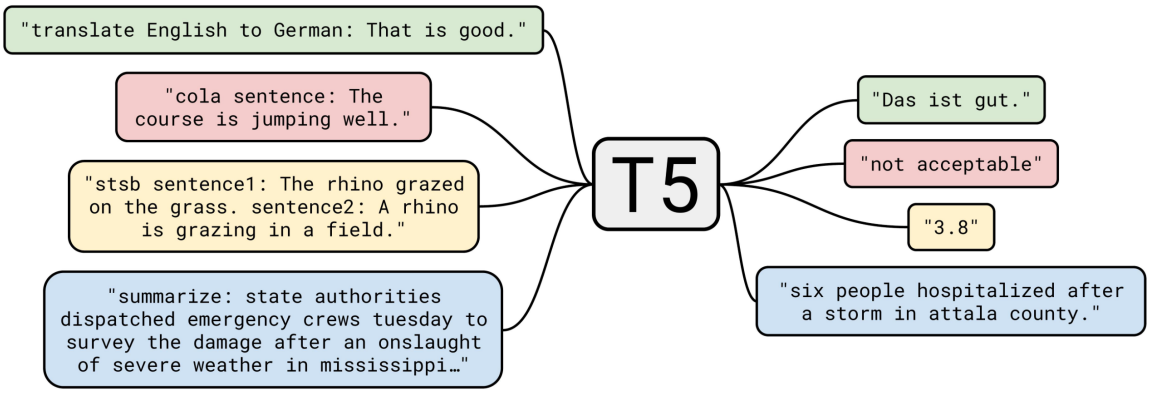

Figure 3: An example of the versatility of T5, and how it will allow us to find errors in general NLP datasets.

```
-----------
i guess feelings aren t meant to be inhibited or prohibited
label: fear
issue confidence: 0.14407547
-----------
i feel unprotected a class post count link href http reprogramming in process
label: sadness
issue confidence: 0.015615896
-----------
i never feel like im not supporting
label: joy
issue confidence: 0.22687584
-----------
i am feeling stressed and more than a bit anxious
label: anger
issue confidence: 0.11214621
-----------
i feel about as helpless and superfluous as i did when jenn had elaine naturally
label: fear
issue confidence: 0.11399045
```

Figure 4: Sentences from the emotion dataset, along with their labels and quality score

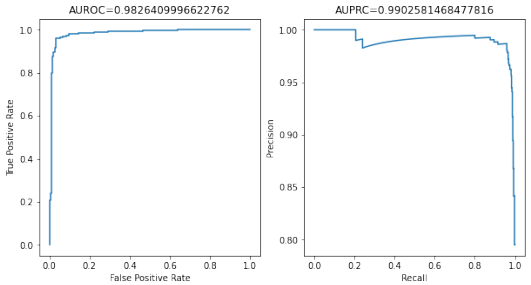

Figure 5: ROC and PRC for SST-2

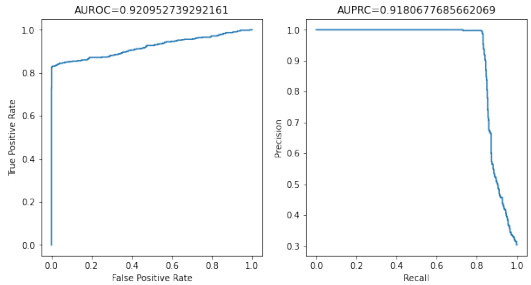

Figure 6: ROC and PRC for emotion

4 Results

For our investigation, we implemented our model pipeline and evaluated it on several datasets, which are summarized in Table 1. We used an emotion classification dataset, a dataset of IMDB sentiments, and the Stanford Sentiment Treebank (SST-2). These are all classification tasks, which our model performed quite well on. We also used a question answering dataset, a summarization dataset, and xsum to test our model's ability to generalize to larger dimensional outputs. The results of our approach on these datasets are summarized in Table 1.

We see from the examples shown in Figure 4 that some of the sentences in the emotion dataset identified as errors were indeed mislabeled, such as "i am feeling stressed and more than a bit anx-

Dataset	AUROC	AUPRC
Emotion	0.92	0.91
IMDB	0.84	0.67
SST-2	0.98	0.99
CNN/Dailymail	0.54	0.34
XSum	0.56	0.37
SQuAD	0.53	0.31

Table 1: Error detection rate of finetuned T5 on datasets with synthetic errors introduced. Datasets were first filtered to have only score of greater than 99%, after which 30% of labels were randomly swapped.

ious", which should be labeled as fear or "i feel unprotected a class post count link href http reprogramming in process", which arguably should not be in the dataset. However, we also note that some correctly labeled samples have low quality scores and were not filtered out.

The results of our study on the T5 model demonstrate both its strengths and limitations in natural language processing (NLP) tasks. On the one hand, T5 achieved excellent results on classification tasks, with an accuracy greater than 90% on all fine-tuned classification tasks. This indicates that T5 is able to accurately identify the correct category or label for a given input with high reliability.

On the other hand, T5 demonstrated poor performance on general NLP tasks such as language translation and question answering. In these tasks, the model made a number of errors and struggled to accurately understand and produce coherent output.

One potential reason for this discrepancy in performance is the nature of the datasets used in the study. Many classification datasets are carefully curated and contain well-defined categories and labels, which may be easier for the T5 model to learn and predict. In contrast, general NLP tasks often involve more complex and open-ended language use, which may be more challenging for the model to accurately process and generate.

Furthermore, our method relies on the distributions of the predicted probabilities to determine the "label score". A more reliable way to extract a score is to construct the confident joint, as described in (Northcutt et al., 2019), one can construct the confident joint, Q. The off-diagonal elements of this matrix can be used to determine a likelihood of error. The downside of this method is expensive computation, i.e the constructed joint confident matrix would be of dimension nxn. In large NLP datasets, this would be on the order of million. Overall, these results suggest that while T5 is a powerful and effective tool for certain NLP tasks, it may not be as well suited for more open-ended and general language processing tasks. Further research is needed to better understand the capabilities and limitations of the T5 model in a wider range of NLP contexts.

5 Discussion

Our results show that for multi-class classification problems, our approach was able to identify errors quite well on these types of datasets. Even on larger dimensional tasks for which the complexity of identifying mislabels significantly increases, our approach still identified a reasonable amount of mistakes. While previous work has been done on identifying errors in token-classification (Wang and Mueller, 2022), we have shown that our approach is able to, at least somewhat, generalize to a larger space of NLP tasks.

One unexpected finding was the compute power needed to find errors well. Our initial goal was to create a general, relatively lightweight, error-finding function that would work on any NLP dataset by leveraging the generality of the T5 transformer. However, from our experiments, we saw that the baseline without finetuning had rather unimpressive results. So to be useful, our error-finding function therefore needed to include an initial finetuning step, which is more computationally expensive, but offers much greater performance, significantly outperforming the base model.

By synthetically introducing errors to originally confident labels, we are able to produce at scale subsets of datasets in which we are reasonably certain whether an input has truly been mislabeled, allowing us to quantify how well errors are identified. However, one problem for this method of evaluation may be that the synthetic errors are not necessarily representative of how errors would be in real data. This would affect how much trust

could be placed in our model in practice. Ideally we would manually identify ground-truth errors in a real dataset, but this approach does not scale well and is infeasible given our resources.

Another shortcoming of our approach is that at the end, a human may still be necessary to go through and verify that the identified labels are truly incorrect. We observed that even for multi-class classification datasets, some of the inputs that our model predicted as mislabeled were ambiguous in nature, such as the sentence "It's somewhat clumsy and too lethargically paced – but its story about a mysterious creature with psychic abilities offers a solid build-up, a terrific climax, and some nice chills along the way" being labeled as 'negative' sentiment. This sort of sentence can be interpreted in different ways depending on where the reader places emphasis, and so whether or not to consider them mislabeled is something that the user would need to decide.

For future directions to take this work, one area to explore is the use of other modern transformers. Given the impressive results of the recent released ChatGPT, we suspect that the generality and performance of our model can still be improved. Another direction for future work is to investigate the error scoring system for question answering datasets. An alternative and more robust approach would be to construct the joint distribution matrix and use that to determine label scores instead. However, this is very computationally intensive and does not scale well. Finding a way to estimate or compute this matrix efficiently would greatly improve the generality of our method.

6 Conclusions

This work presents an approach for identifying label errors in natural language processing datasets using the versatility of the T5 model. The T5 model was able to accurately identify label errors in a general dataset of labeled NLP examples after being finetuned on a T5 model, demonstrating the potential for using large-scale language models to improve the quality of NLP datasets. The T5 model has previously been shown to achieve state-of-the-art performance on a variety of natural language understanding tasks.

We developed an approach to label error detection by finetuning the T5 model on a given dataset, and estimating the confidence of the predicted labels from the finetuned model using a method from cleanlab (Northcutt et al., 2019). This approach was tested on multi-class classification problems and larger dimensional tasks, and was able to identify a reasonable number of mistakes on both types of datasets. We evaluated the effectiveness of the approach by introducing errors to originally confident labels, but this method has the limitation that the synthetic errors may not be representative of real errors in data. Additionally, a human may still be needed to verify that the identified labels are truly incorrect, as some inputs may be ambiguous. Future directions for this work include exploring the use of other modern transformers and investigating an error scoring system for question answering datasets.

7 Code for Experiments

We created a GitHub repository containing the notebooks used to perform our experiments so that others may attempt to replicate our results, or use it to clean their datasets.

8 Impact Statement

There are many ways to increase model performance. One can spend man-hours to fine-tune an algorithm, increase computing power or computing time, acquire more data, or resample a model with cross-validation. These methods can be costly in both time or money. As machine learning becomes ubiquitous in industries and research across the world, limitations of these resources can further amplify financial inequalities. Companies with greater capital can generate more advanced models that outperform smaller companies. Cleaning datasets theoretically helps even the playing field as the resources needed to improve model performance are more accessible to everyone indiscriminately.

We must acknowledge that there are some limitations to the efficacy of this method of cleaning datasets. For one, when using our approach to locate low-confidence labels, it may not always be able to detect all errors. This can be concerning if there is something about the errors left behind that is causing them to be undetected, leaving a bias in the remaining dataset. On the other hand, if the model is somehow able to remove all errors, this could actually be somewhat undesirable in certain cases. Having a perfect dataset could lead to over-fitting and other issues, as noise is imperative for the model's robustness. Furthermore, if our

intent is to make a model human-like, imperfections are inherent in humans and therefore would be an important consideration in the model. So, cleaning out errors may not always be favorable. Lastly, as with many NLP tasks, the conclusion of a model can be ambiguous or subjective. For example, while some may say a sentence is a positive sentiment, others may claim it is negative. Despite these limitations, for data scientists looking to clean their NLP datasets without much manual inspection, our solution provides a method that can be made to work for anyone.

References

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2019. Confident learning: Estimating uncertainty in dataset labels.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Wei-Chen Wang and Jonas Mueller. 2022. Detecting label errors in token classification data.