Humboldt-Universität zu Berlin

Chair of Statistics and Data Science

Research Seminar in Data Science

Advances in Probabilistic Machine Learning

Unpacking BERT with Influence Function: A Method for Accurate Error Identification in Natural Language Inference

Author:
Ruyu Dai

Supervisor:
Victor Medina

Date: 02.2023

Abstract

This paper presents a new approach to using influence functions to identify label errors in natural language inference (NLI) datasets. The study highlights the issue of label errors in NLI datasets and their impact on model performance. Influence functions were applied to measure the loss function change with respect to a removal of a single data point. The study showed that label errors are instances with high influence scores, and the approach managed to identify 80% of the label errors. The study also analyzed the influence of weight decay on the experiment's outcomes and the impact of the sample size on the choice of top eigenvalues in the Arnoldi algorithm.

1 Introduction

Natural language inference (NLI) is a sub-field of natural language processing (NLP) that involves the task of determining the relationship between two given sentences, referred to as premise and hypothesis. The relationship between the two sentences can be classified into entailment, contradiction, or naturalness. NLI is an important task as it showcases a model's ability to determine sentence dependencies, which can be used in other NLP tasks such as text summarization, translation, and Q&A (7).

On the other hand, label errors represent a common problem in natural language inference (NLI) datasets due to the inherent ambiguity of language. The same sentence can have multiple valid interpretations, which can result in inconsistencies in labeling (23). Several studies have highlighted the significant impact of label errors on NLI model performance. For example, (24) found that language models can already perform well in ”only-hypothesis” scenarios, suggesting that these models may rely more on label accuracy than the underlying language context, specifically the premise. Despite the adverse effects of label errors on model performance, this issue is often overlooked in current NLI research.

This paper focuses on this research question: **can the explaining method identify label errors?** Influence functions (16), an emerging explanation method, can address this problem. It measures the loss function change with respect to a removal of a single data point. In this context, label errors are therefore instances with high influence score.

It should be noticed, however, the implementation could be very time-consuming. In this report, we applied influence functions using the latest scaling-up techniques to identify label error in the NLI task. As a result, our approach managed to identify 80% of the label error. Finally, we discussed the robustness of the method by analyzing how weight decay influence the influence score.

2 Influence Function

The focus of our work is to apply influence function to detect label error in NLI datasets. In the following chapter, we briefly introduce the concept of influence function in mathematical terms. Furthermore, we address the computation problem by using the `Arnoldi` method. Finally we illustrate how to measure the quality of retrieving mislabelled error.

2.1 Basic ideas

For a given training dataset $\mathcal{D} = \{z_i : (x_i, y_i)\}_{i=1}^{n}$, the objective of the model is to minimize the loss by finding the parameters $\hat{\theta}^*$ that satisfy $\hat{\theta}^* = \operatorname{argmin} \sum_{i=1}^{n} L(z_i, \theta)$, where $L(z, \theta)$ is the model's loss function.

In the context of tracing prediction to each sample contribution, the change in parameters with respect to a specific sample z_i can be mathematically represented as:

$$\hat{\theta}^*_{-z_i} \overset{\text{def}}{=} \operatorname{argmin} \sum_{j=1, j\neq i}^{n} L(z_j, \theta) \tag{1}$$

Retraining the model for each removed data point to obtain the exact change in parameters can be prohibitively time-consuming, especially when the number of training data points is large. As an alternative, one can treat the removal of a data point as a negative upweighting, which is a computationally efficient method.

More specifically, we can treat removal of a given sample as upweight it by $\epsilon = -\frac{1}{n}$, which is close to zero when n is large enough. To obtain the influence of a specific sample z on the loss function, we can apply the chain rule:

$$\begin{aligned} \mathcal{I}_{up,loss}(z, z_{test}) \overset{\text{def}}{=} \frac{dL(z_{test}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\bigg|_{\epsilon=-\frac{1}{n}\approx 0} &= \frac{dL(z_{test}, \hat{\theta}_{\epsilon,z})}{d\hat{\theta}_{\epsilon,z}} \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\bigg|_{\epsilon=-\frac{1}{n}\approx 0} \\ &= -\nabla_\theta L(z_{test}, \hat{\theta}_{-z_i})^T H^{-1}_{\hat{\theta}^*} \nabla_\theta L(z, \hat{\theta}^*), \end{aligned} \tag{2}$$

where we use (9) to calculate $\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\bigg|_{\epsilon=-\frac{1}{n}\approx 0}$

$$\mathcal{I}_{up,params}(z) \overset{\text{def}}{=} \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\bigg|_{\epsilon=0} = -H^{-1}_{\hat{\theta}} \nabla_\theta L(z, \hat{\theta}^*) \tag{3}$$

2.2 use Arnoldi to calculate IF

The computational cost of applying Influence Functions (IF) can be prohibitively high. For instance, in a BERT model comprising approximately 110 million parameters, each represented in double-precision, the exact Hessian matrix would require 96GB of memory, rendering it impossible to store the Hessian matrix during calculation. To address this challenge, (26) proposed a method called `Arnoldi` that combines several strategies to reduce the computation cost.

The initial step of the approximation involves utilizing a promising subset $\mathcal{D}' << \mathcal{D}$, where $\mathcal{D}$ is the original training dataset. The algorithm subsequently performs the remaining computations on this sub-dataset to reduce the computational burden.

The second step entails converting the approximated Hessian matrix into a diagonal form. This involves two stages. The first stage employs the Arnoldi iteration to obtain the eigenvalues w_i of the Hessian on the sub-dataset. To simplify the matrix further, the last rows of the Hessian matrix and the last w_i are discarded. Next, this matrix is distilled to its top $\tilde{p}$ eigenvalues λ_i' with the corresponding eigenvectors e_i'. The final projection matrix G represents the projection of the corresponding eigenvectors e_i' in the w_i basis. The goal of this step is to simplify the H matrix into a diagonalized form, allowing the calculation of Equation 2 to be reduced to a simple dot product of the gradient.

Finally, Calculate 3 as dot products of $G \cdot \nabla_\Theta L(z_{test})\hat{H}^{-1}$ and $G \cdot \nabla_\Theta L(z)$, where $\hat{H}$ is a diagonalized matrix $\{\lambda_i\}_{i=1}^{\tilde{p}}$.

2.3 evaluation of the influence function performance

As noted by (16), instances with high self-influence scores can indicate data outliers. To evaluate the retrieval quality, we deliberately flip 20% of the labels in the test data and calculate each instance's influence score $\mathcal{I}_{x,x}$. The evaluation is performed based on the area under the curve (AUC) and average precision (AP) metrics.

3 Experiment Set-up

The training and test data used in this study are obtained from the Multi-Genre Natural Language Inference (MNLI) dataset (29), which includes 393k premise and hypothesis pairs spanning three relations across 10 different genres. Specifically, we utilize the same 10,000 training samples selected in (13). Furthermore, to simplify the task, we transform the original contradiction and neutralness labels to non-entailment, resulting in a dichotomous label scheme. For the test dataset, we randomly select 50 samples from the corresponding test dataset in MNLI

Since the training samples are identical, we can readily employ the fine-tuned BERT model from (13). The model achieves an accuracy of 70% on the newly selected test dataset. The experiments are conducted using Colab and TPU v2, with the area under the curve (AUC) and average precision (AP) metrics employed to evaluate the model's performance on the proposed experiment.

4 Results and Discussion

4.1 Experiment Results

In this study, 20% of the test data was deliberately corrupted, meaning it was manually transformed into label-error data. The objective was to investigate this hypothesis: **high influence score can indicate label error accurately.**

$\tilde{p}$	**AUC**	**AP**
1	57.99	32.84
5	**78.47**	**46.22**
10	78.47	46.22

Table 1: Retrieving mislabeled test on 50 test samples from MNLI test-dev.

The results, presented in Table 1, reveal that the `Arnoldi` algorithm performs optimally with a number of projectors $\tilde{p} = 5$, yielding an AUC of 78.47% and an AP of 46.22%. Our findings indicate that the best performance is achieved at a relatively small value of $\tilde{p}$, which may be attributed to the small size of the test dataset. In the case of incorrect model predictions, a small test dataset can lead to a more substantial percentage drop in accuracy. To account for this limitation, we repeat the experiment on a slightly larger test dataset while utilizing the same fine-tuned BERT model.

After increasing the test sample, we witness the expected improvement of retrieval quality. However, this improvement is relatively small compared to the increase of the test data size.

$\tilde{p}$	**AUC**	**AP**
1	59.02	33.79
5	80.77	47.42
8	**81.25**	**48.23**
10	81.25	48.23

Table 2: Retrieving mislabeled test on 200 test samples from MNLI test-dev.

4.2 Discussion on the influence of weight decay

During the experiments, we noticed a significant influence of weight decay. It is a common regularization technique in neural networks to avoid model overfitting. This method penalizes the sum of square of coefficient weight. In the NLP field, l2 regularization is more popular used because its easier to calculate the gradient on the loss function from the l2 regularization.

How weight decay affects the IF performance w.r.t. specific model is not clear yet. On the one hand, it could enhance interpretability by preventing the to-be-interpreted model from overfitting, leading to a clearer distinction between corrupted and correct labels. On the other hand, weight decay may lead to a larger gap in Taylor's approximation when applying IF to calculate instance influence. Specifically, treating the removal of a sample as up-weighting it by $\epsilon = -\frac{1}{n}$ results in the approximation of the change in parameters through Taylor's expansion, as shown in Equation 4.

$$\hat{\theta}_{\epsilon,z} - \hat{\theta}^{*} = \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon} \cdot \epsilon \tag{4}$$

To investigate the effect of weight decay on the experimental outcomes, we vary the weight decay parameter using three different values: 1×10^{-3}, 5×10^{-2}, and 1×10^{-3}, and then replicate the experiment.

As shown in Figure 1, excessive weight decay results in a significant drop in the performance of the influence function. In this case, the AUC and AP are both below 50%. Conversely, when weight decay is too low, indicating model underfitting, the graph shows a peculiar trend: as the number of projections increases, the accuracy decreases. This phenomenon is perplexing, as including more eigenvalues in the Hessian matrix approximation should result in a more precise calculation of the influence function. This indicates that the model may be considering additional samples as data outliers, instead of just the ones with flipped labels. In conculsion, we found that an appropriate setting of weight decay is important to the application of influence function in this task.

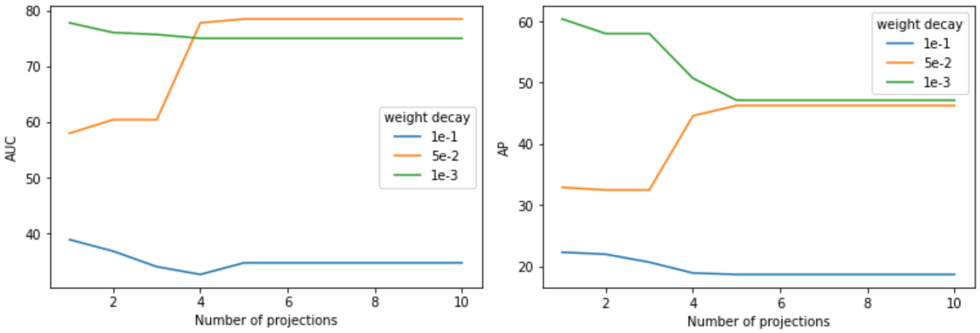

Figure 1: AUC and AP for retrieval of mislabeled MNLI examples as a function of $\tilde{p}$ with different weight decay for BERT

5 Related Work

5.1 Applying influence function in label error detection

The selection of interpretation methods should be based on the downstream task at hand. In our case, we aim to utilize the method to identify label errors, which pertains to the broader domain of data cleaning. Our focus is on the traceability of the explanatory method, which is not a ubiquitous characteristic of explainable artificial intelligence (XAI) in general. Under the context of error detection, out-of-distribution detection (OOD) (27, 19, 12) and label error detection (8, 22, 10) are two non-explainable methods that are often compared to our proposed approach.

Compared with these methods, recent studies suggest that influence functions are more sensitive to model hyperparameter settings (6, 5). To overcome this limitation, an improved contrast-based influence function for label error identification was proposed in (18). Another drawback of using influence function is its computational burden. Despite researchers' efforts to develop speed-up techniques for Hessian approximation (25, 26), its $\mathcal{O}(p^2)$ complexity makes it challenging to apply to complex deep neural networks, where p is the size of the model parameters.

5.2 Explanation methods in other tasks

Model fairness Above, we discussed our method's application in identifying label errors, which refers to a small portion of mislabeled data in the dataset. In contrast, the following section addresses situations where data is correctly labeled but shows systematic bias. As noted by (14), the design choices made in NLP models can magnify social biases, while the use of automated data processing may result in unexpected and systematic harm. In other words, researchers are turning to XAI to mitigate systematic bias in language models and ensure that they are fair and unbiased.

A well-established approach in the field of interpretability is to identify local feature attribution to address the issue of systematic bias in language models. For example, in (28), the authors employed the SHAP method to determine the most critical features or words in the model's decision-making process and assess the degree to which politically biased data affects the model's performance. Similarly, (4) used a combination of feature importance analysis, along with local and global explanations, to identify the necessary and sufficient conditions for a text to be classified as hate speech. However, the intersection of model fairness and interpretability is still largely limited to hate recognition. As highlighted by (3), several challenges remain in this area, such as the generalization problem of local interpretations, dependence on human detection, susceptibility to fair-washing, and striking a balance between outcome fairness and procedural fairness.

Revealing model learning mechanism Finally, interpretation methods can also be utilized to reveal a model's learning mechanisms and understand adversarial attacks. For instance, (13) utilized influence functions to quantify the degree to which a model relies on superficial syntactic properties (21), specifically lexical overlap. Furthermore, (17) proposed `TransSHAP`, an extension of SHAP that explains BERT predictions in tweet sentiment analysis. Another notable contribution is the order-sensitive SHAP method introduced by (20), which highlights how language models rely excessively on spurious correlations between particular words and the label, rather than meaningful semantic relationships. By doing so, the proposed method can aid in identifying vulnerabilities that may be exploited by adversarial attacks.

Nevertheless, it is noteworthy that several researchers(1, 2) have pointed out the inconsistency of popular interpretation methods in revealing language model mechanisms. In their analysis of various NLP tasks, (15) compared attention-based methods to other feature importance measures and convincingly demonstrated that attention does not provide meaningful explanations.

6 Conclusion

In this study, we conducted comprehensive research on the application of interpretability methods in NLP. In particular, we introduce a novel approach to using influence functions to identify label errors in the natural language inference (NLI) dataset. Our findings reveal that the choice of top eigenvalues in the `Arnoldi` algorithm and its corresponding best performance are influenced by the sample size. Moreover, we demonstrate that the choice of weight decay has a significant impact on the experiment's outcomes. To further advance this field, future research may focus on developing techniques to improve the computational efficiency of influence function calculations and enhance their robustness to hyperparameter settings. Ultimately, we anticipate that this sample-based interpretation method can provide users with quick and reliable insights into language model predictions and enable them to identify deficiencies in the training dataset.

Appendices

A Litrature review of popular explaining methods

Methodology	Explanation	Application
LIME	token level	hate recognition(4)
Gradient-based saliency score	token level	revealing learning mechanism (13, 15)
Attention-based	token level	revealing learning mechanism (15)
SHAP	mostly token level[1]	hate speech recognition(4, 28), revealing learning mechanism (17)
Influence Function	example level	detecting label error (18), unveiling data artifacts (13)

[1] (11) introduced Data Shapley, a method to quantify individual data contribution to the whole algorithm. The explanation granularity is finer than token level but less than example level.

B A sanity check

The integration of Influence Functions (IF) into large neural networks has yet to attain the status of a well-established practice among the scientific community(16). This study endeavors to demonstrate the validity of the IF method by adopting the methodology used in (13). The hypothesis is that the elimination of the most positive influential instances will result in a decline in prediction accuracy, while the elimination of the most negative influential instances will result in an improvement in accuracy. Conversely, a randomly selected removal of samples is expected to result in a decline in accuracy, although to a lesser degree. The removal of least influential samples is predicted to produce similar results.

The procedure involves the fine-tuning of the model, the calculation of Arnoldi IF for each testing sample, and the identification of the top 10% positive and negative influential training samples and the 10% least influential samples. The performance of the model is then evaluated through the removal of each set of influential training samples, repeated 5 times to mitigate variance.

The findings of the study affirm the anticipated trends and validate the utilization of IF as a tool for quantifying the explanation of the BERT model. The elimination of the top 10% positive influential samples led to a pronounced decrease in prediction

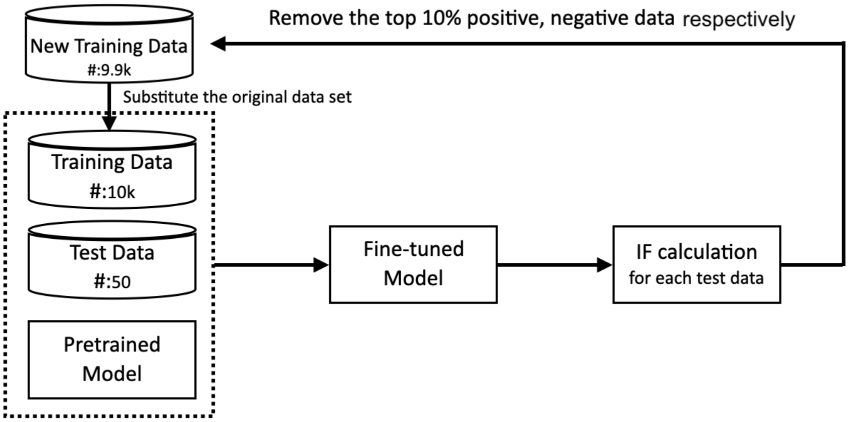

Figure 2: Procedure for a sanity check

Removal type	**Avg. Δ in prediction confidence**
Positively influential	-5.68% ($\pm 3.13\%$)
Negative influential	$+1.47\%$ ($\pm 2.48\%$)
Least influential	$+0.92\%$ ($\pm 1.02\%$)
Random	$+0.28\%$ ($\pm 1.45\%$)

Table 3: Sanity check for BERT using influence function

accuracy. However, the elimination of the other forms of influential instances did not produce substantial results. This can be attributed to the structural resemblance between the test and training datasets, which limited the influence on the predictions.

References

[1] Arjun Reddy Akula and Song-Chun Zhu. Attention cannot be an explanation. *ArXiv*, abs/2201.11194, 2022.

[2] Bing Bai, Jian Liang, Guan Zhang, Hao Li, Kun Bai, and Fei Wang. Why attentions may not be interpretable? *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2020.

[3] Esma Balkir, Svetlana Kiritchenko, Isar Nejadgholi, and Kathleen C. Fraser. Challenges in applying explainability methods to improve the fairness of nlp models, 2022.

[4] Esma Balkir, Isar Nejadgholi, Kathleen C. Fraser, and Svetlana Kiritchenko. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection, 2022.

[5] Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training examples via relative influence, 2020.

[6] Samyadeep Basu, Phillip E. Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *ArXiv*, abs/2006.14651, 2020.

[7] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference, 2015.

[8] Derek Chong, Jenny Hong, and Christopher D. Manning. Detecting label errors by using pre-trained language models, 2022.

[9] R. Dennis Cook and Sanford Weisberg. Residuals and influence in regression. 1982.

[10] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning, 2019.

[11] Amirata Ghorbani and James Y. Zou. Data shapley: Equitable valuation of data for machine learning. *ArXiv*, abs/1904.02868, 2019.

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

[13] Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[14] Sara Hooker. Moving beyond "algorithmic bias is a data problem". *Patterns*, 2, 2021.

[15] Sarthak Jain and Byron C. Wallace. Attention is not explanation, 2019.

[16] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *ArXiv*, abs/1703.04730, 2017.

[17] Enja Kokalj, Blaž Škrlj, Nada Lavrač, Senja Pollak, and Marko Robnik-Šikonja. BERT meets shapley: Extending SHAP explanations to transformer-based classifiers. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 16–21, Online, April 2021. Association for Computational Linguistics.

[18] Faisal Ladhak, Esin Durmus, and Tatsunori Hashimoto. Tracing and removing data errors in natural language generation datasets, 2022.

[19] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2017.

[20] Kaiji Lu and Anupam Datta. Order-sensitive shapley values for evaluating conceptual soundness of nlp models, 2022.

[21] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics.

[22] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. 2021.

[23] Ellie Pavlick and Tom Kwiatkowski. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694, 2019.

[24] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference, 2018.

[25] Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating training data influence by tracing gradient descent, 2020.

[26] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *AAAI Conference on Artificial Intelligence*, 2021.

[27] Koh Pang Wei, Sagawa Shiori, Marklund Henrik, Xie Sang Michael, Zhang Marvin, Balsubramani Akshay, Hu Weihua, Yasunaga Michihiro, Phillips Richard Lanas, Gao Irena, Lee Tony, David Etienne, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S.

Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts, 2020.

[28] Maximilian Wich, Jan Bauer, and Georg Groh. Impact of politically biased data on hate speech classification. In *Workshop on Abusive Language Online*, 2020.

[29] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2017.