Cascading Explaining - A universal method to improve NLP model

Guo Zikun

2021225559

Department of Artificial Intelligence, Kyungpook national University, Republic of Korea

gzk798412226@gmail.com

**Abstract—Recently some interpretable methods have been proposed, among which the interpretation method for question answering systems: Integrated Gradients. It is a more popular method. We propose a novel method to improve the performance of the model based on the characteristics of the Integrated Gradients applied in the question answering system: Cascading Explaining. This method is to extract information from the interpretation of the model output, and then let the model predict again. We have used some experiments to prove that this method has successfully passed the sanity check, and has greatly improved the accuracy and probability of model prediction.** **At the same time, this is a general form method: it is not limited by models and interpretation methods. In other words, the performance of any question answering system can be improved by this method.**

**Keywords— Explainable ai, Question Answering System, Integrated Gradients, Sanity Check**

# INTRODUCTION

As machine learning grows in complexity and impact, much hope rests on explanation methods as tools to elucidate important aspects of learned models. Explanations could potentially help satisfy regulatory requirements, help practitioners debug their model, and perhaps, reveal bias or other unintended effects learned by a model. Integrated Gradients[1] is an increasingly popular tool designed to highlight relevant features in the input. Although this method is compared with other methods such as LIME[2] and SHAP[3], it shows its uniqueness and innovation. But still explaining the precious hard work of the machine learning model faces a methodological challenge: the difficulty of evaluating the scope and quality of the model interpretation. The lack of principled guidelines confuses practitioners when making decisions. Understanding the input and output behavior of a deep network gives us the ability to improve it. This intelligibility is for all computer programs, including machine learning models. Attribution has other applications. They can be used in machine learning-driven products to provide recommended rationale. For example, a deep network based on imaging can help inform the doctor about the part of the image that led to the recommendation. This can help doctors understand and compensate for a person's strengths and weaknesses model. Developers can also use attribution in an exploratory sense. For example, we can use deep networks to extract insights that can be used in rule-based systems.

Our contributions

* We propose a specific, easy-to-implement test to improve model performance : Cascading Explaining Universal method with no limitation by models and interpretation methods
* We show that the interpretation of the bert-based question answering system is not independent of the model. Therefore, to explain this type of model is meaningful and easy to understand

# METHOD

A. Cascading Explaining

The basic goal of Cascading Explaining is well understood. This method is inspired by the attention mechanism of the human brain: when we are doing reading comprehension, we will first focus on the entire article and read it through, and then make a choice based on the goal in our mind: which paragraph may hide what we need Answer. So we read again in the new natural paragraph until we find the answer we hope.

Cascading Explaining is also same attention mechanism of the human brain. Given an input to the model, the contribution of different elements to the model output is calculated through the gradient integration algorithm. Then the high contribution value often hides the answer. Delete or replace the unimportant with meaningless annotations such as delimiters or paragraphs that have no effect on the model, and then re-input this processed sentence into the model. The model will Output the same answer, but the probability of a ground truth has been increased.

In order to ensure that the correct answer is not deleted or replaced. Cascade interpretation was proposed. In the process of processing model interpretation, there may be wrong answers, but because the interpretation is independent of the input, after multiple interpretations of the first range, we make changes to the later based on the interpretation of the initial model and the initial input After the data are classified and compared. Normally, the correct answer is usually accompanied by a possibility. The higher the possibility, the more certain the model is. When the model gives a low probability answer, we need to wonder whether the model really understands the current data. This is the essence of Cascading Explaining. It is precisely because of multiple local adjustments that the misinterpretation caused by mispredictions is reduced.

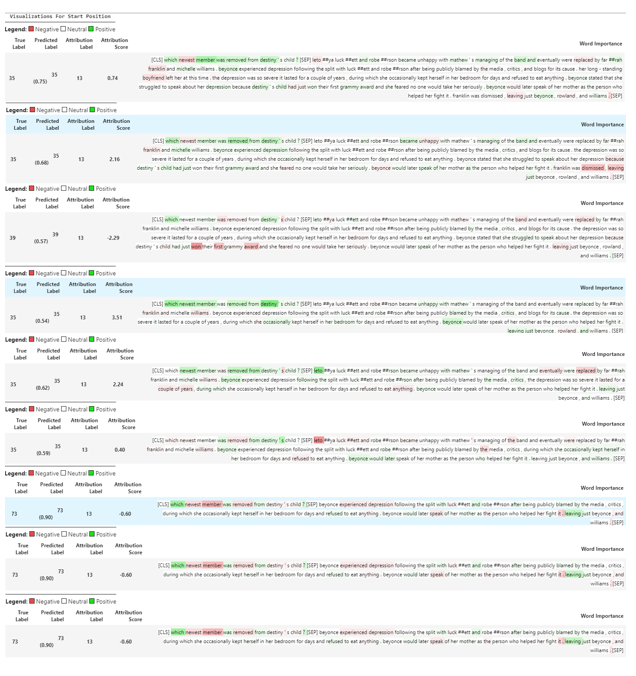


Fig. 1 Example of an Cascading Explaining, This example only provides the first 9 treatments, After deleting the tokenziers with low contribution to the result, different interpretations are obtained, and then iteratively processed, and finally the initial input text is extracted into a short sentence. This sentence is the sentence containing the answer. According to Figure 2, it can also be easily found The shortest model will provide higher possibilities.

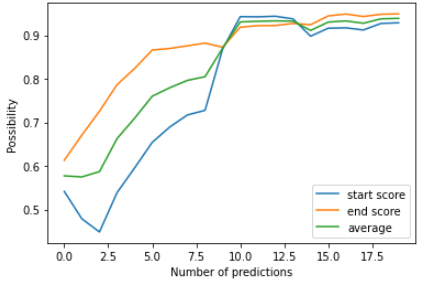


Fig. 2 The X-axis corresponds to the number of loop predictions, and the y-axis uses the probability of the answer. This curve clearly shows that the processed input will get a higher probability.

B. Sanity Check

In our formal setup, an input is a vector x ∈ R d . A model describes a function S : R d → R C , where C is the number of classes in the classification problem. An explanation method provides an explanation map E : R d → R d that maps inputs to objects of the same shape. We now briefly describe some of the explanation methods we examine. The supplementary materials contain an in-depth overview of these methods. Our goal is not to exhaustively evaluate all prior explanation methods, but rather to highlight how our methods apply to several cases of interest. The gradient explanation for an input x is Egrad(x) = ∂S ∂x [22, 23, 8]. The gradient quantifies how much a change in each input dimension would a change the predictions S(x) in a small neighborhood around the input.

The data randomization test compares a given saliency method applied to a model trained on a labeled data set with the method applied to the same model architecture but trained on a copy of the data set in which we randomly permuted all labels. If a saliency method depends on the labeling of the data, we should again expect its outputs to differ significantly in the two cases. An insensitivity to the permuted labels, however, reveals that the method does not depend on the relationship between instances and labels that exists in the original data. Speaking more broadly, any explanation method admits a set of invariances, i.e., transformations of data and model that do not change the output of the method. If we discover an invariance that is incompatible with the requirements of the task at hand, we can safely reject the method. As such, our tests can be thought of as sanity checks[4] to perform before deploying a method in practice.

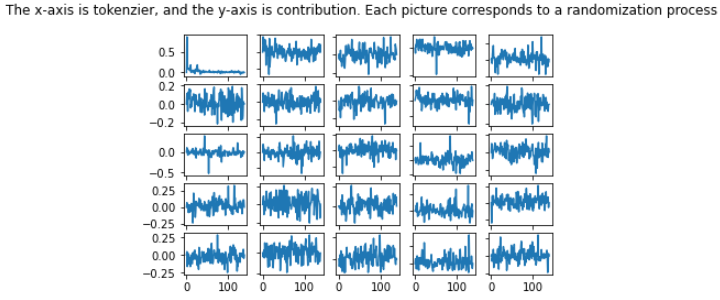


Fig. 3 Cascading Randomization randomize the weights of a model starting from the top layer, successively, all the way to the bottom layer. Then plot the contribution value of the obtained tokenizer.

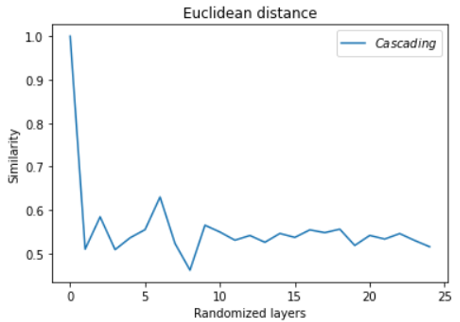


Fig. 4 Calculate different random layers and calculate the contribution of each layer's tokenizer, use Euclidean distance to calculate the similarity of the contribution.

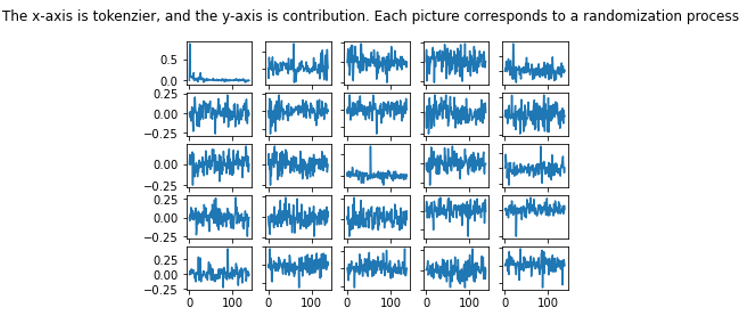


Fig. 5 Independent Randomization the conduct an independent layer-by-layer randomization with the goal of isolating the dependence of the explanations by layer.

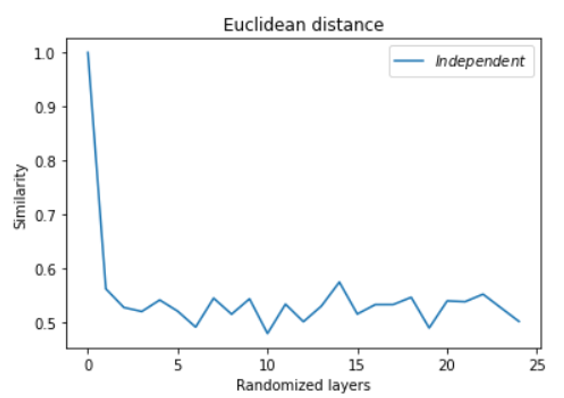


Fig. 6 Independent Random layers and calculate the contribution of each layer's tokenzer, use Euclidean distance to calculate the similarity of the contribution.

Base on the similarity of contribution show that the transformer model can pass the sanity check test, means the explanation is useful and meaningful in the cascading explain method

# REFERENCES

1. Mukund Sundararajan, Ankur, Taly Qiqi Yan “Axiomatic Attribution for Deep Networks” , arXiv:1703.01365, 13 Jun 2017
2. Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”, arXiv:1602.04938, 16 Feb 2016
3. Scott Lundberg, Su-In Lee , “A Unified Approach to Interpreting Model Predictions” , To appear in NIPS 2017 Mon, 22 May
4. Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, Been Kim , “Sanity Checks for Saliency Maps”, Computer Vision and Pattern Recognition, Mon, 8 Oct 2018