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

# RELATED WORK

Information retrieval (IR) systems [4] have served as the standard baseline for QA tasks [5]. However, the lack of lexical overlap in many QA datasets between questions and answers [6], makes standard IR approaches that rely on strict lexical matching less applicable. Several IR systems have been modified to use distributional similarity to align query terms to the most similar document term for various tasks, including document matching [7], short text similarity [8], and answer selection [9]. However, using only a single most similar term can lead to spurious matches, e.g., with different word senses. Here we expand on this by allowing a one-to-many mapping between a question term and similar answer terms to better represent how on-context a given answer candidate is. Negative information has also been show to be useful in answer sentence selection [10]. We also include negative information

Integrated Gradients identify two fundamental axioms—Sensitivity and Implementation Invariance that attribution methods ought to satisfy. Use axioms to guide the design of a new attribute method called integrated gradient. This method does not need to modify the original network and is very simple to implement; it only requires a few calls to the standard gradient operator. Apply this method to a couple of image models, several text models and a chemical model to demonstrate its ability to debug the network, extract rules from the network, and enable users to better interact with the model.

Formally, suppose we have a function *F* : Rn → [0, 1] that represents a deep network. Specifically, let x ∈ Rn be the input at hand, and x0 ∈ Rn be the baseline input. For image networks, the baseline could be the black image, while for text models it could be the zero embedding vector. We consider the straightline path (in Rn) from the baseline x0 to the input x, and compute the gradients at all points along the path. Integrated gradients are obtained by cumulating these gradients. Specifically, integrated gradients are defined as the path intergral of the gradients along the straightline path from the baseline x 0 to the input x. The integrated gradient along the *ith* dimension for an input x and baseline x 0 is defined as follows. Here, ∂F (x) is the gradient of F(x) along the *ith* dimension. For most deep networks, it is possible to choose a baseline such that the prediction at the baseline is near zero (*F*(x 0 ) ≈ 0). (For image models, the black image baseline indeed satisfies this property.) In such cases, there is an intepretation of the resulting attributions that ignores the baseline and amounts to distributing the output to the individual input features.

# 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.

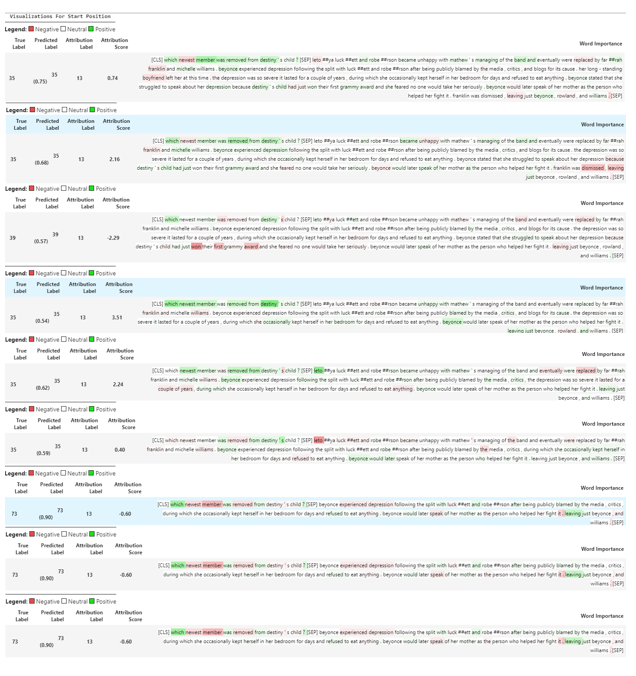


Fig. 1 Example of an Cascading Explaining, This example only provides the first 9 treatments, After deleting the tokenziers with low contribution to the result, different interpretations are obtained, and then iteratively processed, and finally the initial input text is extracted into a short sentence. 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.

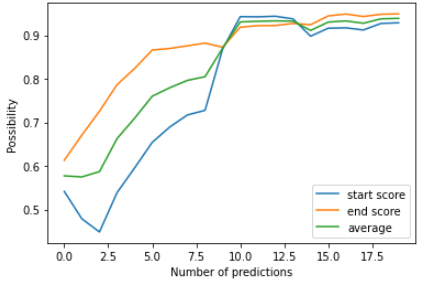


Fig. 2 The X-axis corresponds to the number of loop predictions, and the y-axis uses the probability of the answer. This curve clearly shows that the processed input will get a higher probability.

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[10] to perform before deploying a method in practice.

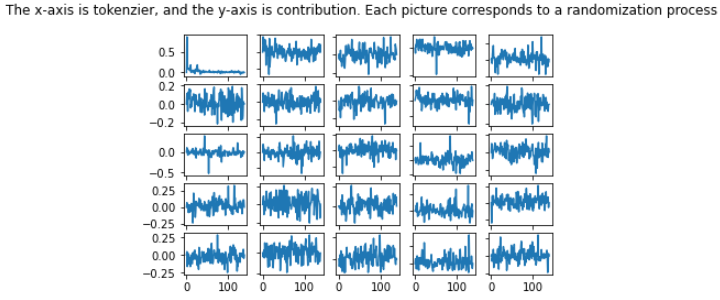


Fig. 3 Cascading Randomization randomize the weights of a model starting from the top layer, successively, all the way to the bottom layer. Then plot the contribution value of the obtained tokenizer.

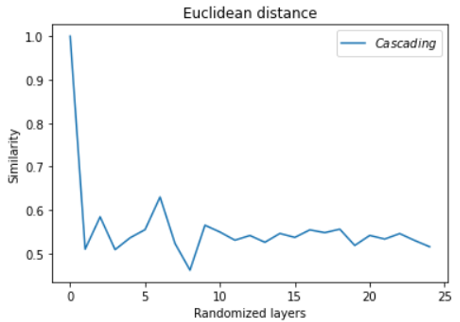


Fig. 4 Calculate different random layers and calculate the contribution of each layer's tokenizer, use Euclidean distance to calculate the similarity of the contribution.

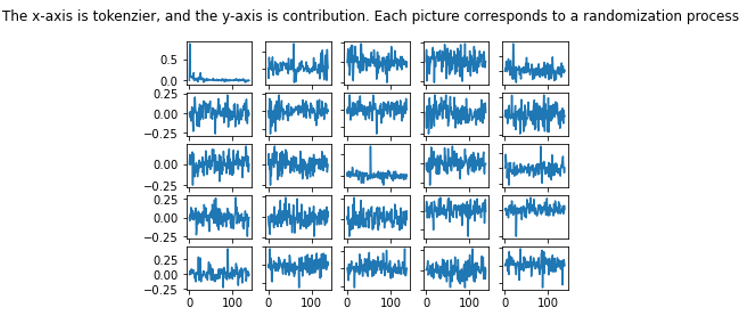


Fig. 5 Independent Randomization the conduct an independent layer-by-layer randomization with the goal of isolating the dependence of the explanations by layer.

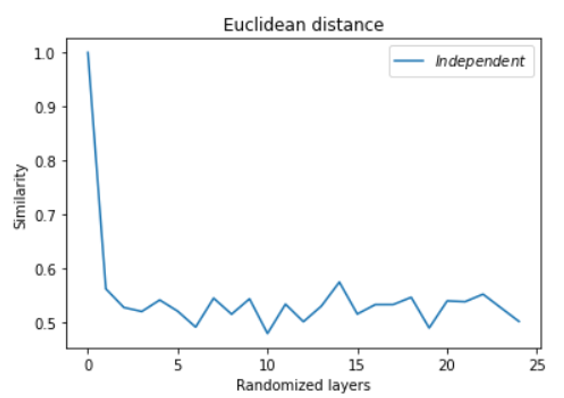


Fig. 6 Independent Random layers and calculate the contribution of each layer's tokenzer, use Euclidean distance to calculate the similarity of the contribution.

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.

# CONCLUSION

We have introduced a quick, simple but powerful interpretation of the QA method using a pre-trained model to continuously verify itself between the question and the answer. Despite its simplicity, our method is suitable for complex systems and is a general-purpose framework for improving model prediction probabilities.

# FUTURE DICTIONS

This general framework can only improve the probability at present. We have found some situations in the experiment that can make the model prediction wrong, after continuous self-explanation two-point verification, and finally make the answer correct. This is also Our ultimate goal. In the future, we will choose other methods to characterize such features and standardize them at the same time, so that we can really make the model perform self-checking and verify whether our answer is correct or not through the method of interpretation.

# REFERENCES

1. Mukund Sundararajan, Ankur, Taly Qiqi Yan “Axiomatic Attribution for Deep Networks” , ACM:1703.01365, 13 Jun 2017
2. Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”, ACM: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. Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to Rank Answers to Non-Factoid Questions from Web Collections. Computational Linguistics 37, 2 (2011), 351–383.
5. Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval 3, 4 (2009), 333–389
6. Tom Kenter and Maarten De Rijke. 2015. Short text similarity with word embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 1411–1420.
7. Daniel Fried, Peter Jansen, Gustave Hahn-Powell, Mihai Surdeanu, and Peter Clark. 2015. Higher-order Lexical Semantic Models for Non-factoid Answer Reranking. Transactions of the Association for Computational Linguistics 3 (2015), 197–210. Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. arXiv preprint arXiv:1803.05457 (2018)
8. Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 1532–1543.
9. Rishav Chakravarti, Jiri Navratil, and Cicero Nogueira dos Santos. 2017. Improved Answer Selection with Pre-Trained Word Embeddings. arXiv preprint arXiv:1708.04326 (2017).
10. 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