Explanatory and Actionable Debugging for Machine Learning: A TableQA Demonstration

Minseok Cho, Gyeongbok Lee, Seung-won Hwang
Yonsei University, Seoul, Korea
{whatjr,alias_n,seungwonh}@yonsei.ac.kr

ABSTRACT

Question answering from tables (TableQA) extracting answers from tables from the question given in natural language, has been actively studied. Existing models have been trained and evaluated mostly with respect to answer accuracy using public benchmark datasets such as WikiSQL. The goal of this demonstration is to show a debugging tool for such models, explaining answers to humans, known as explanatory debugging. Our key distinction is making it "actionable" to allow users to directly correct models upon explanation. Specifically, our tool surfaces annotation and models errors for users to correct, and provides actionable insights.

ACM Reference Format:
Minseok Cho, Gyeongbok Lee, Seung-won Hwang. 2019. Explanatory and Actionable Debugging for Machine Learning: A TableQA Demonstration. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331404

1 INTRODUCTION

Question answering (QA) models, answering natural language questions from various sources such as text (TextQA) or tables (TableQA), have been actively studied, powered by shared efforts on large-scale public datasets for training and evaluation. In both tasks, state-of-the-art models have exceeded human performance in terms of accuracy, though recent work questions right answers from such model can be "lucky guesses" and thus easily perturbed [9].

Inspired, we question a classic pipeline of optimizing for accuracy metric alone. Instead, visual analytic community proposed explanatory debugging tools [5, 14, 15] enabling to observe model behaviors, not only from decision accuracy, but also from the following richer context:

- Decision with training samples: Machine decision can be connected to related training instances.
- Decision with model internals: Machine decision can be explained by model internals.
- Counterfactual testing: Users can alter testing cases for "what if" explorations of model behaviors.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331404

Query: Sum all SB less than 14. **Answer:** 14

Player	Team	3B	HR	RBI	BB	SO	SB	CS
Altuve, J	HOU	4	24	81	58	84	32	6
Blackmon, C	COL	14	37	104	65	135	14	10
Garcia, A	CWS	5	18	80	33	111	5	3
Murphy, D	WSH	3	23	93	52	77	2	0
Turner, J	LAD	0	21	71	59	56	7	1

Figure 1: Example of cell attention supervision

However, though existing tools help users to observe model behaviors in new angles, they do not focus on empowering users to correct when they think otherwise, or analytics are not yet **actionable**: Actionability is a key factor in explanatory debugging [5], as users ignore explanations when they cannot benefit by acting on them.

In contrast, we focus on actionability and propose three new goals, adding actionability on top of the above three distinctions. Further, we will present empirical evidences that such human intervention contributes to accuracy increases. Specifically, we identify our three new goals.

- **[G1] Actionable annotation editor:** Our editor connects a training instance where machine decision disagrees with ground truth. Training data can be edited when *machine decision is right, but the ground truth is wrong*. We investigated WikiSQL in this way and found 37% of annotations for scalar aggregation is incorrect in this set, and our editor supports an easy change of annotations.
- **[G2] Actionable model editor:** Explanatory debugging surfaces model misbehaviors, even when *both machine decision and annotation are correct*, or "lucky guesses". Meanwhile, accuracy metric cannot detect such error, leading to brittle models to adversarial perturbations. While existing tools focus on showing attention for understanding behaviors, our tool explores **cell attention supervision**, where users can simply change attention to retrain models. Figure 1 shows example user edits on which cells to attend.
- **[G3] Counterfactual testing:** Lastly, training instances may not be challenging enough to ensure the robustness of the model: Our editor allows users to test adversarial training cases, which can be augmented as adversarial training resources.

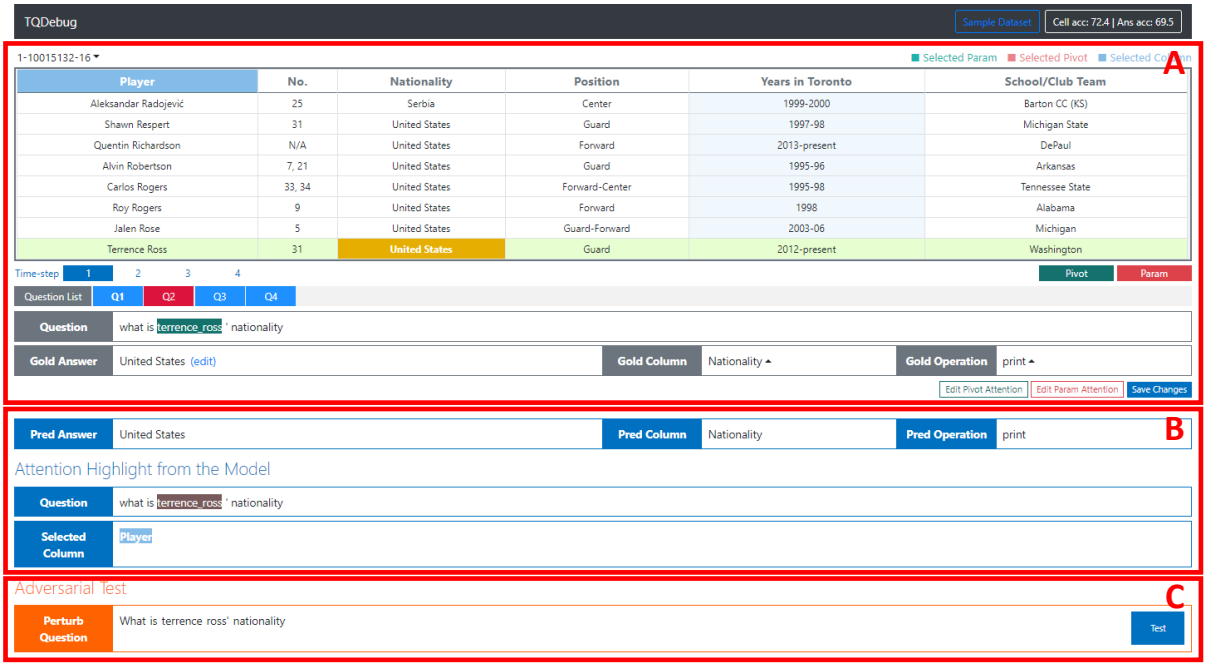

Figure 2: Snapshot of TQDebug Framework

2 RELATED WORK

Explanatory debugging was pioneered in [5] for a simple text classifier, and similar explanatory tools have been demonstrated for language and QA tasks [1, 6, 13, 15]. Counterfactual explorations or attention modifications were also demonstrated, mostly for the purpose of understanding models in "what if" scenarios [8, 14]. However, whether such changes contribute to an increase in accuracy was left unexplored, to the best of our knowledge.

Our key difference is actionability. We connect machine decision with testing instance and model internal, so that users can act on unexpected behaviors to fix them. Our editor empowers users to fix annotation mistakes, inaccurate attention, and/or training instances that are too easy. These changes reflect model changes,by retraining. For explainability in more general domains, refer to discussions in [11, 12].

One of the enabling techniques for actionablity is *attention supervision* [3, 7]. where attention weight that is close to human perception is observed to be more effective for end-tasks. In other word, models can be retrained to more closely reflect human perception of attention, which leads to accuracy gains as well. Our work validated the effectiveness of attention supervision as actionable insights for TableQA debugging: Specifically, we adopt a state-of-the-art in TableQA with attention supervision [2], and show human intervention can improve this model even further in Table 1.

3 DEMONSTRATION SCENARIOS

Figure 2 thee main modules (marked as 'A', 'B', and 'C'), for achieving G1, G2, and G3 discussed in Section 1. We elaborate each module in the next sections respectively.

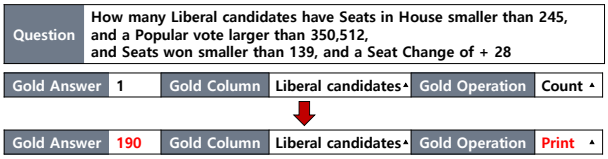

Figure 3: Annotation editor

3.1 G1: Annotation Editor

Annotation Editor provides an environment for fixing wrong annotations. Figure 3 shows an actual example instance of WikiSQL, where the operation annotated by the user does not agree with that predicted from the model. For the question illustrated in this example, the annotation of operation which should be used to get the correct answer is wrong, which can be corrected by human intervention. This question asks to print the cell value of 'Liberal candidates' which meet the conditions of the question, so operation should be corrected to 'print' operation, instead of 'count' as annotated.

Using the tool, we found 37% of scalar aggregation in WikiSQL is inaccurately labeled, for which our editing tool enables easy changes. Models can be retrained after such changes. after which models can be retrained for correction.

In addition to answer labels, users can use this tool to modify annotations such as cells which should be considered to derive the final answer. Corrected datasets lead to higher accuracy after re-training, as we show later in Table 1.

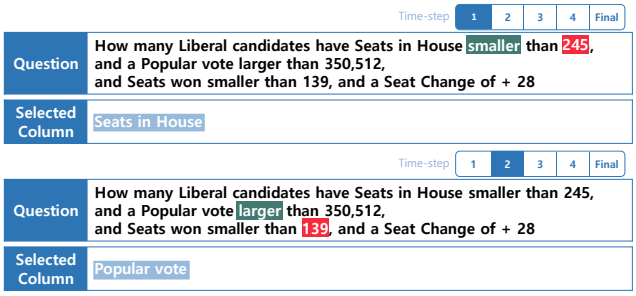

Figure 4: Visualization for attention weight

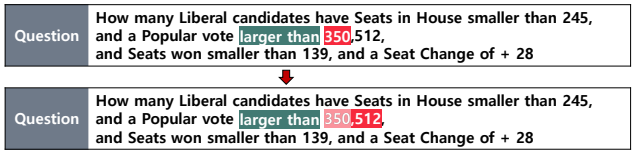

Figure 5: Editor for attention supervision

3.2 G2: Text-Cell Attention Supervision

Visual representation for attention weight on data cells on the table or on question words shows whether right words are highlighted for understanding questions or finding answers. By showing attention weights at each timestep, it can help users to understand how the model predicts the answer.

Figure 4 shows words from the question and column from the table, which the model focuses on at each timestep (i.e. $t = 1, 2$). In this figure, question words and column with highest attention weight are highlighted, such that users can look at the case and reweigh the attention to correct models by modifying the label.

In addition, as shown in Figure 5, our attention editor allows user to modify attention labels, by moving or dragging the highlighted box to another word in the question that users would rather highlight. Attention label shown in red suggests that the model is focusing on '350' to find an answer, which is only a substring of an actual number, '350,512'. Our editor enables users to enlarge the attention span as in the figure below. With such attention supervision using our editor, users can retrain models to better fit user understanding of the task.

3.3 G3: Adversarial Paraphrases

As we overviewed, many existing QA models are weak to adversarial examples. To enhance robustness, users can edit the questions to perform the adversarial test for each instance. For example, the given question can be edited to its paraphrases using different wordings, to check whether the model relies too much on exact matches.

Figure 6 shows an example of the result of adversarial test with perturbed question. In this example, user changes the phrase from 'What's' to 'What is', and the changed question preserves the semantic of original question. If the model prediction changes after semantic-preserving changes, we can call them *adversarial paraphrase*. The model gets reportedly more robust, when (re-)trained

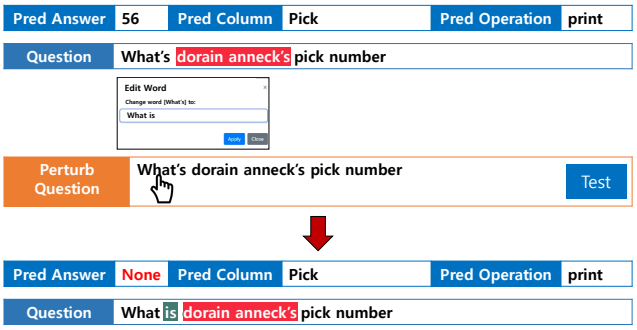

Figure 6: Editor for adversarial paraphrase

Table 1: Performance before and after using editor

Model	*CellAcc*	*AnswerAcc*
w/o editor	60.9	59.5
w/ editor	**77.2 (+26.8%)**	**75.3 (+26.6%)**

with augmented training resources from such diversified paraphrases [4, 10].

4 EFFECTIVENESS RESULTS

We empirically study whether achieving three goals contribute to performance gain.

(1) **For G1**, we randomly sample 1k queries with a scalar answer, from which we find 37% are incorrectly annotated using our tool. Figure 7 shows wrong annotations we find. For questions starting with 'how many', in general, the correct answer can be obtained from using 'print' or 'count' operation, while most annotations suggest 'min' or 'max'. Our tool enables to retrain the model based on corrected annotations from user intervention.

(2) **For G2**, we compare with a state-of-the-art model with attention supervision: WikiOPs [1] is an attention supervision dataset, automatically extracted from WikiSQL dataset [16], by transforming the SQL statements into cell supervisions. Figure 5 is an attention editor we built on top of this model, where users can add text and cell supervisions or correct. First, for quantitative evaluation, Table 1 compares the effectiveness of attention editor, by comparing with WikiOPs in terms of the following two metrics: cell accuracy and answer accuracy– The former evaluates whether the model selects the cells correctly to get the correct answer, and answer accuracy evaluates whether the model predicts the correct answer. The model with our editor outperforms the model trained on original WikiOPs . Second, Figure 8 shows qualitative evidences that our editor contributing to the model change behaviors as expected.

(3) **For G3**, our tool not only provides the environment for adversarial test, but also allows the user to retrain the model with dataset augmented with adversarial training resources for enhancing the robustness of the model. Figure 9 shows

[1] https://github.com/MinseokCho/NeuralOperator

Question	Gold Op	Correct Op
[How many ~]		
How many kids go to Brechin High School?	max	print
How many number of lakes are there in the Valdez-Cordova (CA) area?	min	count
How many Inhabitants were there after 2009 in the Municipality with a Party of union for trentino?	average	print
[What is the ~]		
What is the attendance in the game in the 2010-11 season, when the score was 2–0?	min	print
What is the highest rank of a player from Ethiopia?	max	min
What is the population (2010 census) if s barangay is 51?	count	print
[Others]		
With less than 7 Silver medals, how many Gold medals did Canada receive?	max	sum
During which round was the first defensive end with an overall rank larger than 186 drafted?	max	print
What year is center Ann Wauters?	sum	print

Figure 7: Illustration of wrong annotations from WikiSQL

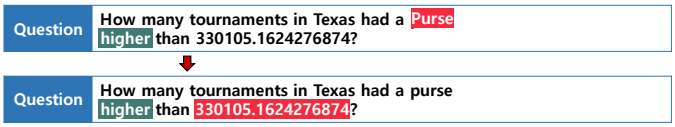

Figure 8: Change in attention weight before and after using editor

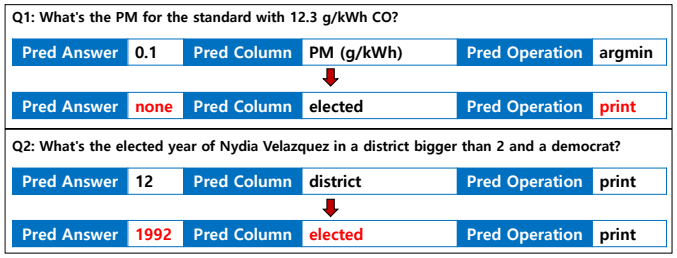

Figure 9: Change in results from augmenting adversarial training resources

the change in results of the model retrained on the dataset added adversarial training resources. From Q1 and Q2, we can see that the model often makes a mistake when the question starts with "what's", for which we can augment adversarial paraphrases. As shown in Figure 9, the retrained model does not show the same weakness.

5 CONCLUSION

This demo revisits a classic pipeline for TableQA where model errors are observed as accuracy metric alone. Model developers have responded to errors by adding training data or rewriting models. In contrast, we show an explanatory tool showing model behavior, which is actionable, such that users can directly edit wrong annotations or model behaviors. Another contribution is to show empirical evidences that user intervention invited from actionable tools, contribute to the increase in model accuracy.

6 ACKNOWLEDGEMENT

This work was supported by IITP grant funded by the Korea government(MSIT) (No.2017-0-01779, XAI). Hwang is a corresponding author.

REFERENCES

[1] 2018. AllenNLP Reading Comprehension Demo. In *http://demo.allennlp.org/machine-comprehension*.
[2] Minseok Cho, Reinald Kim Amplayo, Seung-won Hwang, and Jonghyuck Park. 2018. Adversarial TableQA: Attention Supervision for Question Answering on Tables. In *ACML*.
[3] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. 2017. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding* (2017).
[4] Seung-won Hwang and Kevin Chang. 2007. Probe minimization by schedule optimization: Supporting top-k queries with expensive predicates. In *IEEE TKDE*.
[5] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. *International Conference on Intelligent User Interfaces, Proceedings IUI* 2015. https://doi.org/10.1145/2678025.2701399
[6] Gyeongbok Lee, Sungdong Kim, and Seung won Hwang. 2019. QADiver: Interactive Framework for Diagnosing QA Models. In *AAAI*.
[7] Chenxi Liu, Junhua Mao, Fei Sha, and Alan L Yuille. 2017. Attention Correctness in Neural Image Captioning.. In *AAAI*.
[8] S. Liu, Z. Li, T. Li, V. Srikumar, V. Pascucci, and P. Bremer. 2018. NLIZE: A Perturbation-Driven Visual Interrogation Tool for Analyzing and Interpreting Natural Language Inference Models. *IEEE Transactions on Visualization and Computer Graphics* (2018).
[9] Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic Answer Networks for Machine Reading Comprehension. *arXiv preprint arXiv:1712.03556* (2017).
[10] Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase Diversification using Counterfactual Debiasing. In *AAAI*.
[11] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *KDD*. 1135–1144.
[12] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI*.
[13] Andreas Rücklé and Iryna Gurevych. 2017. End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights. In *ACL*.
[14] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. 2019. Seq2Seq-Vis: A Visual Debugging Tool for Sequence-to-Sequence Models. In *IEEE TVCG*.
[15] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M. Rush. 2018. LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks. In *IEEE TVCG*.
[16] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR* abs/1709.00103 (2017).