Towards Robust Numerical Question Answering: Diagnosing Numerical Capabilities of NLP Systems

Jialiang Xu[1], Mengyu Zhou[2], Xinyi He[3], Shi Han[2], Dongmei Zhang[2]
[1] University of Illinois at Urbana-Champaign [2] Microsoft Research [3] Xi'an Jiaotong University
jx17@illinois.edu, hxyhxy@stu.xjtu.edu.cn, {mezho, shihan, dongmeiz}@microsoft.com

Abstract

TL;DR

- The DNC framework diagnoses numerical capability weaknesses in Natural Language Processing systems in a systematic way.

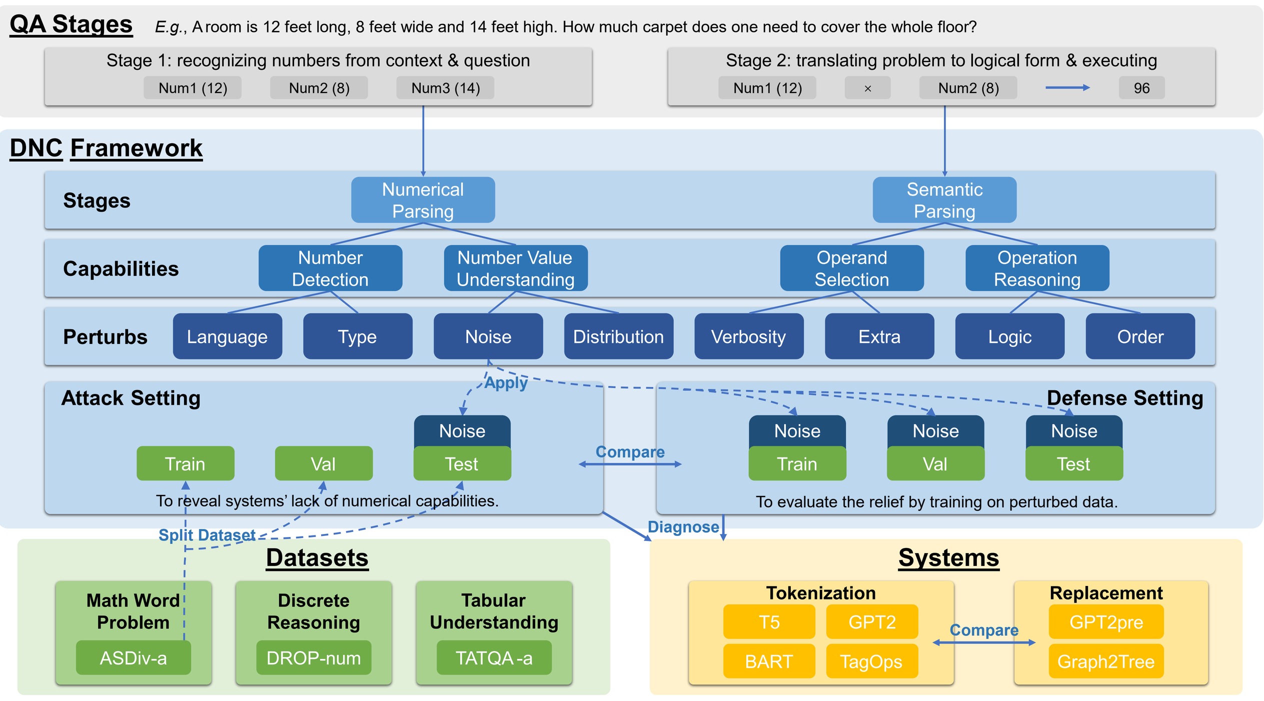
Longer Version

- **Task:** The DNC (**D**iagnosing **N**umerical **C**apability) framework is proposed to probe the robustness in systems on Question Answering datasets that require numerical reasoning capabilities [1] [2].
- **Methodology:** Four numerical capabilities, stemming from the two solving stages of numerical QA questions, are highlighted. Accordingly, eight perturbations are designed to probe these capabilities. Being trivial to humans, these perturbations are expected not to affect the system performance significantly.
- **Results:** Empirical Results show that current systems are error-prone on the perturbed test set ("Attack"), and demonstrate non-trivial performance drop even trained on the perturbed training set ("Defense").

DNC Framework

Overview

- Two **stages**: Numerical Parsing, Semantic Parsing
- Four **capabilities**: Numerical Detection, Number Value Understanding, Operand Selection, Operation Reasoning
- Eight **perturbations**: Language, Type, Noise, Distribution, Verbosity, Extra, Logic, Order
- Two **settings**: Attack, Defense



Overview of DNC Framework. The process of Numerical QA solving is divided into two logical stages. Four capabilities are required to complete the stages, each maps to two perturbations. Perturbations can be applied to appropriate train / validation / test splits of Numerical QA datasets under Attack or Defense Setting. Models of the NLP systems are trained and then evaluated on the perturbed datasets as a diagnosis of their numerical capabilities.

Perturbation Examples

- Orgnaized into groups of capabilities
- For each of the perturbations, a case is shown where the perturbation is applied to a sample numerical QA problem, and T5 failed to induce correct answer after the perturbation.

Capability	Perturbation	Example Problem Pair	T5 Prediction
Number Detection	Language	Original: A mailman has to give out 192 pieces of junk mail. If he goes to 4 blocks, how many pieces of junk mail should he give each block? Perturbed: A mailman has to give out one hundred and ninety-two pieces of junk mail. If he goes to four blocks, how many pieces of junk mail should he give each block?	Original: 192 / 4 ✓ Perturbed: 92 / 4 × Expected: 192 / 4
	Type	Original: There were 105 parents in the program and 698 pupils, too. How many people were present in the program? Perturbed: There were 105.0 parents in the program and 698.0 pupils, too. How many people were present in the program?	Original: 105 + 698 ✓ Perturbed: 105 + 688 × Expected: 105 + 698
Number Value Understanding	Noise	Original: Tony had $20. He paid $8 for a ticket to a baseball game. At the game, he bought a hot dog for $3. What amount of money did Tony have then? Perturbed: Tony had $20.2. He paid $8.5 for a ticket to a baseball game. At the game, he bought a hot dog for $3.5. What amount of money did Tony have then?	Original: 20 - 8 - 3 ✓ Perturbed: 208.52 - 3.5 × Expected: 20.2 - 8.5 - 3.5
	Distribution	Original: Frank had $16. After buying some new toys he had $8 left.How much did he spend on toys? Perturbed: Frank had $1281. After buying some new toys he had $478 left.How much did he spend on toys?	Original: 16 - 8 ✓ Perturbed: 1215 - 878 × Expected: 1281 - 478
Operand Selection	Extra	Original: John has twelve shirts. Later he bought four more shirts. How many shirts does John have in total? Perturbed: John has twelve shirts. Later he bought four more shirts. Frank had $16. How many shirts does John have in total?	Original: 12 + 4 ✓ Perturbed: 16 + 12 × Expected: 12 + 4
	Verbosity	Original: The roller coaster at the state fair costs 6 tickets per ride. If 8 friends were going to ride the roller coaster, how many tickets would they need? Perturbed: The roller coaster at the state fair costs 6 (not 30) tickets per ride. If 8 (not 119) friends were going to ride the roller coaster, how many tickets would they need?	Original: 6 * 8 ✓ Perturbed: 8 * 119 × Expected: 6 * 8
Operation Reasoning	Logic	Original: Jack received 8 emails in the morning and 2 emails in the afternoon. How many emails did Jack receive in the day? Perturbed: Jack received 8 emails in the morning and 2 emails in the afternoon. How many more emails did Jack receive in the morning than in the afternoon?	Original: 8 + 2 ✓ Perturbed: 8 + 2 × Expected: 8 - 2
	Order	Original: A DVD book holds 126 DVDs. There are 81 DVDs already in the book. How many more DVDs can be put in the book? Perturbed: There are 81 DVDs already in the book. A DVD book holds 126 DVDs. How many more DVDs can be put in the book?	Original: 126 - 81 ✓ Perturbed: 81 - 126 × Expected: 126 - 81

Examples of DNC Perturbations and Corresponding Predictions by T5. For each perturbation an example original and perturbed problem pair is shown. The rightmost column shows some error cases where T5 generates correct equation on the original problem but fails on the perturbed. The ground truth equation of the perturbed problem is also provided after "Expected".

Formalization

Expected Behavior

- When a perturbation does not confuse humans, it should not confuse a robust numerical QA systems either. I.e.,

$$f: (P, B) \rightarrow T \Leftrightarrow f: (P^*, B^*) \rightarrow T^*$$

where f is a learned numerical QA system, P, B, and T are the prompt, the body, and the target of the numerical QA problem, respectively. The asterisk $(\cdot)^*$ denotes the perturbed version of the corresponding element.

Observed Discrepancy

- Empirical results demonstrate a discrepancy between the Expected and the actual behavior.
- Attack: systems trained on original data fails on perturbed.

$$f: (P_{train}, B_{train}) \rightarrow T_{train} \nRightarrow f: (P^*_{test}, B^*_{test}) \rightarrow T^*_{test}$$

- Defense: systems trained on perturbed data fails on perturbed.

$$f: (P^*_{train}, B^*_{train}) \rightarrow T^*_{train} \nRightarrow f: (P^*_{test}, B^*_{test}) \rightarrow T^*_{test}$$

- Thus, the systems are inferred to possess numerical capability weaknesses and have been picking up spurious correlations.

Experiments

Experiment Settings

- **Attack**: to construct a challenge set to evaluate the corresponding numerical capability of existing systems
- **Defense**: to investigate to the extent to which performance drops can be alleviated by using the perturbations as a data augmentation approach.

Setting	Attack	Defense
Train on	train	train*
Validate on	val	val*
Test on	test*	test*

Dataset and Systems

- In this paper, we used ASDiv-a [3], DROP [4], and TATQA [5] as our Numerical Question Answering datasets. For DROP and TATQA, we filtered out DROP-num and TATQA-a, the numerical subsets of them. The statistics of these datasets are shown below.

Dataset	# Training	# Validation	# Testing
ASDiv-a	974	122	122
DROP-num	42258	5282	5283
TATQA-a	1971	245	247

- We selected representative systems on each dataset and test their performance against perturbations. For the ASDiv-a dataset, we use Graph2Tree [6]. For the DROP dataset, we use BART-base and T5-base from Huggingface. For the TATQA dataset, we utilize TagOps with the RoBERTa backbone as described in the original paper.

Experiment Result

Configuration		ASDiv-a								DROP-num		TATQA -a
		T5		BART		GPT2		Graph2Tree		T5	BART	TagOps
Setting	Perturbation	Acc_{eq}	Acc_{ans}	Acc_{eq}	Acc_{ans}	Acc_{eq}	Acc_{ans}	Acc_{eq}	Acc_{ans}	Acc	Acc	Acc
Attack (Δ)	Language	-18.85%	-18.85%	-23.77%	-27.05%	-12.30%	-12.30%	-7.65%	-7.38%	-10.62%	-14.73%	-18.62%
	Type	-37.70%	-11.48%	-32.79%	-15.57%	-17.21%	-10.66%	0.27%	1.09%	-7.70%	-11.06%	-5.34%
	Noise	-36.89%	-36.89%	-18.85%	-21.31%	-9.84%	-9.02%	0.27%	0.55%	-	-	-
	Distribution	-16.39%	-14.75%	-29.51%	-18.03%	-13.11%	-13.11%	-6.56%	-6.56%	-	-	-
	Verbosity	-41.80%	-44.26%	-25.41%	-29.51%	-10.66%	-11.48%	-33.33%	-33.88%	-9.58%	-13.31%	-1.90%
	Extra	-25.41%	-27.87%	-41.80%	-45.90%	-28.69%	-28.69%	-53.83%	-54.64%	-11.79%	-11.67%	-1.21%
	Logic	-29.51%	-27.87%	-36.89%	-35.25%	-25.41%	-23.77%	-28.42%	-21.86%	-	-	-14.29%
	Order	-34.43%	-5.74%	-33.61%	-4.10%	-27.87%	-7.38%	-33.33%	-7.10%	-	-	1.12%
Defense (Δ)	Language	-12.30%	-13.93%	-19.67%	-24.59%	2.46%	2.46%	-7.65%	-7.38%	0.07%	-1.84%	-7.59%
	Type	-11.48%	-12.30%	-4.92%	-6.56%	3.28%	4.10%	1.64%	1.91%	0.46%	-0.95%	2.93%
	Noise	-14.75%	-14.75%	-3.28%	-4.92%	3.28%	4.10%	0.55%	0.27%	-	-	-
	Distribution	-20.49%	-20.49%	-8.20%	-9.84%	-8.20%	-9.02%	-6.83%	-6.01%	-	-	-
	Verbosity	-15.57%	-16.39%	-5.74%	-7.38%	-0.82%	0.00%	-0.27%	1.09%	-5.13%	-1.84%	2.25%
	Extra	0.00%	1.64%	-2.46%	-4.10%	-17.21%	-18.03%	-20.22%	-17.76%	-11.32%	-10.44%	-9.14%
	Logic	-	-	-	-	-	-	-	-	-	-	13.64%
	Order	-25.41%	-4.10%	-27.87%	-7.38%	-1.64%	23.77%	-29.23%	-7.92%	-	-	19.47%
Original	None	68.03%	72.95%	67.21%	72.95%	44.26%	45.08%	66.94%	68.58%	49.42%	50.36%	42.41%

The Results of DNC Framework. Five NLP systems are evaluated with three Numerical QA tasks under both Attack and Defense settings. The symbol "Δ" stands for the absolute metric difference between the current setting and the original setting. The color scale represents the distance from the original setting, deeper means further from the original setting. For ASDiv-a, Acc_{eq} and Acc_{ans} refer to the prediction accuracy of ground truth equations and denotation accuracy of answers, respectively. For DROP-num and TATQA-a, Acc refers to the denotation accuracy of the answers.

Insights

- **Attack**: 1) most systems experienced significant performance drop. 2) Between the two DNC goals, Semantic Parsing causes a more severe challenge. 3) Among the considered systems, Transformer-based Seq2Seq systems are more sensitive than the tasks-specific Graph2Tree system against the perturbations stemming from the Numerical Parsing goal.
- **Defense**: 1) the mechanism helps alleviate systems' lack of corresponding numerical capabilities. 2) The lack according to Semantic Parsing gets more recovery. 3) Among the considered systems, Transformer-based Seq2Seq systems benefits more from Defense.

Experiment Summary

- It is demonstrated that severe numerical weaknesses exist in current Numerical QA systems ("Attack"), and they can not be trivially eliminated via, although benefiting from, an automatic data augmentation process ("Defense").
- The systems' weaknesses are explicitly profiled in a quantitative and interpretable manner through the models' susceptibility difference to a diversity of perturbations.

Relevant Future Directions

- **Target**: Logical Form Generation *vs.* Answer Predicting. Logical Form Generation is when systems generate the logical form which is later input to external symbolic executing systems. Answer Predicting is when systems directly predict the output answer in an end-to-end manner.
- **Numbers**: Tokenization *vs.* Replacement. Tokenization divides numbers into potentially multiple sub-word level tokens. Replacement substitutes numbers with special tokens in the input which are later re-substituted with the original number in the output logical forms.

References & Info

[1] Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2020a. The gap of semantic parsing: A survey on automatic math word problem solvers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(9):2287–2305.
[2] Yihuai Lan, Lei Wang, Qiyuan Zhang, Yunshi Lan, Bing Tian Dai, Yan Wang, Dongxiang Zhang, and Ee-Peng Lim. 2022. Mwptoolkit: An open-source framework for deep learning-based math word problem solvers. Proceedings of the AAAI Conference on Artificial Intelligence, 36(11):13188–13190.
[3] Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 975–984, Online. Association for Computational Linguistics.
[4] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
[5] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3277–3287, Online. Association for Computational Linguistics.
[6] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2080–2094, Online. Association for Computational Linguistics.

Jialiang Xu
jx17@illinois.edu

Personal Page
I'm applying for grad school starting Fall 2023!

Project Page
Code, Data, Video

ILLINOIS Microsoft 西安交通大学 XI'AN JIAOTONG UNIVERSITY EMNLP 2022