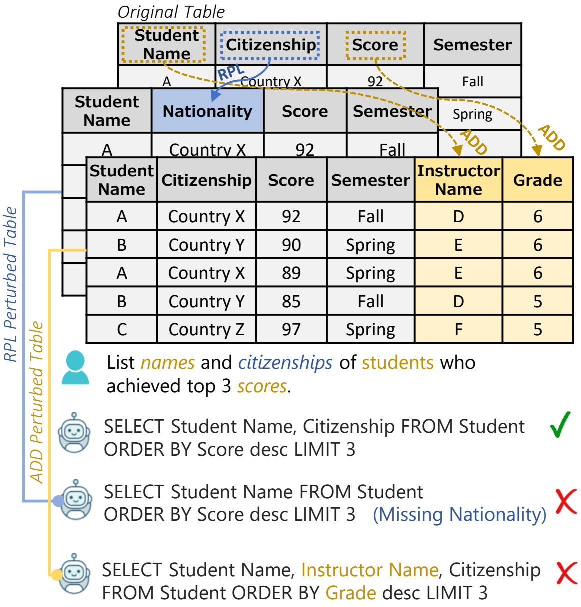
Towards Robustness of Text-to-SQL Models Against Natural and Realistic Adversarial Table Perturbation

Xinyu Pi[1*], **Bing Wang**[2*], **Yan Gao**[3], **Jiaqi Guo**[4], **Zhoujun Li**[2], **Jian-Guang Lou**[3]
[1] University of Illinois Urbana-Champaign, Urbana, USA,
[2] State Key Lab of Software Development Environment, Beihang University
[3] Microsoft Research Asia
[4] Xi'an Jiaotong University, Xi'an, China
`xinyupi2@illinois.edu; {bingwang, lizj}@buaa.edu.cn`
`jasperguo2013@stu.xjtu.edu.cn; {yan.gao, jlou}@microsoft.com`

Abstract

The robustness of Text-to-SQL parsers against adversarial perturbations plays a crucial role in delivering highly reliable applications. Previous studies along this line primarily focused on perturbations in the natural language question side, neglecting the variability of tables. Motivated by this, we propose the **A**dversarial **T**able **P**erturbation (**ATP**) as a new attacking paradigm to measure the robustness of Text-to-SQL models. Following this proposition, we curate ADVETA, the first robustness evaluation benchmark featuring natural and realistic ATPs. All tested state-of-the-art models experience dramatic performance drops on ADVETA, revealing models' vulnerability in real-world practices. To defend against ATP, we build a systematic adversarial training example generation framework tailored for better contextualization of tabular data. Experiments show that our approach not only brings the best robustness improvement against table-side perturbations but also substantially empowers models against NL-side perturbations. We release our benchmark and code at: https://github.com/microsoft/ContextualSP.



Student Name	Citizenship	Score	Semester	Instructor Name	Grade
A	Country X	92	Fall	D	6
B	Country Y	90	Spring	E	6
A	Country X	89	Spring	E	6
B	Country Y	85	Fall	D	5
C	Country Z	97	Spring	F	5

List *names* and *citizenships* of students who achieved top 3 *scores*.

SELECT Student Name, Citizenship FROM Student ORDER BY Score desc LIMIT 3 ✓

SELECT Student Name FROM Student ORDER BY Score desc LIMIT 3 (Missing Nationality) ✗

SELECT Student Name, Instructor Name, Citizenship FROM Student ORDER BY Grade desc LIMIT 3 ✗

Figure 1: Adversarial examples based on table perturbations for a Text-to-SQL parser. Leaving the NL question unchanged, both replacement of column names (e.g., replace "Citizenship" with "Nationality") and addition of associated columns (e.g., add "Instructor Name" based on "Student Name"; add "Grade" based on "Score") mislead the parser to incorrect predictions.

1 Introduction

The goal of Text-to-SQL is to generate an executable SQL query given a natural language (NL) question and corresponding tables as inputs. By helping non-experts interact with ever-growing databases, this task has many potential applications in real life, thereby receiving considerable interest from both industry and academia (Li and Jagadish, 2016; Zhong et al., 2017; Affolter et al., 2019).

Recently, existing Text-to-SQL parsers have been found vulnerable to perturbations in NL questions (Gan et al., 2021; Zeng et al., 2020; Deng et al., 2021). For example, Deng et al. (2021) removed the explicit mentions of database items in a question while keeping its meaning unchanged, and observed a significant performance drop of a Text-to-SQL parser. Gan et al. (2021) also observed a dramatic performance drop when the schema-related tokens in questions are replaced with synonyms. They investigated both multi-annotations for schema items and adversarial training to improve parsers' robustness against permutations in NL questions. However, previous works only studied the robustness of parsers from the perspective of NL questions, neglecting variability from the other side of parser input – tables.

We argue that a reliable parser should also be robust against table-side perturbations since they are inevitably modified in the human-machine interaction process. In business scenarios, table main-

[*] Equal contributions during the internship at Microsoft Research Asia.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2007 - 2022
May 22-27, 2022 ©2022 Association for Computational Linguistics

tainers may *(i)* rename columns due to business demands and user preferences. *(ii)* add new columns into existing tables when business demands change. Consequently, the extra lexicon diversity introduced by such modifications could harm performances of unrobust Text-to-SQL parsers. To formalize these scenarios, we propose a new attacking paradigm, **A**dversarial **T**able **P**erturbation (**ATP**), to measure parsers' robustness against *natural and realistic* ATPs. In accordance with the two scenarios above, we consider both **RE**P**L**ACE (**RPL**) and **ADD** perturbations in this work. Figure 1 conveys an intuitive understanding of ATP.

Ideally, ATP should be conducted based on two criteria: *(i)* Human experts consistently write correct SQL queries before and after table perturbations, yet parsers fail; *(ii)* Perturbed tables look natural and grammatical, and are free from breakage of human language conventions. Accordingly, we carefully design principles for RPL/ADD and manually curate the **ADVE**rsarial **T**able perturb**A**tion (**ADVETA**) benchmark based on **three** existing datasets. All evaluated state-of-the-art Text-to-SQL models experience drastic performance drops on ADVETA: On ADVETA-RPL, the average relative percentage drop is as high as 53.1%, whereas on ADVETA-ADD is 25.6%, revealing models' lack of robustness against ATPs.

Empirically, model robustness can be improved by adversarial training, i.e. re-train models with training set augmented with adversarial examples (Jin et al., 2020). However, due to the different natures of structured tables and unstructured text, well-established text adversarial example generation approaches are not readily applicable. Motivated by this, we propose an effective **C**ontextualized **T**able **A**ugmentation (CTA) approach that better leverages tabular context information and carry out ablation analysis. To summarize, the contributions of our work are three-fold:

- To the best of our knowledge, we are the *first* to propose definitions and principles of **A**dversarial **T**able **P**erturbation (**ATP**) as a new attacking paradigm for Text-to-SQL.

- We contribute **ADVETA**, the first benchmark to evaluate the robustness of Text-to-SQL models. *Significant performance drops* of state-of-the-art models reveals that there is much more to explore beyond high leaderboard scores.

- We design **CTA**, a systematic adversarial training example generation framework tailored for better contextualization of tabular data. Experiments show that our approach brings model *best robustness gain* and *lowest original performance loss*, compared with various baselines. Moreover, we show that adversarial robustness brought by CTA generalizes well to NL-side perturbations.

2 Adversarial Table Perturbation

We propose the **A**dversarial **T**able **P**erturbation (**ATP**) paradigm to measure robustness of Text-to-SQL models. For an input table and its associated NL questions, the goal of ATP is to fool Text-to-SQL parsers by perturbing tables naturally and realistically. More specifically, human SQL experts can consistently maintain their correct translations from NL questions to SQL with their understanding of language and table context. Formally, ATP consists of two approaches: **RE**P**L**ACE (**RPL**) and **ADD**. In the rest of this section, we first discuss our consideration of table context, then introduce conduction principles of RPL and ADD.

2.1 Table Context

Tables consist of explicit and implicit elements – both are necessary for understanding table context. "Explicit elements" refer to table captions, columns, and cell values. "Implicit elements", in our consideration, contains **T**able **P**rimary **E**ntity (**TPE**) and domain. (Relational) Tables are structured data recording domain-specific attributes (columns) around some central entities (TPE) (Sumathi and Esakkirajan, 2007). Without the explicit annotation, humans could still make correct guesses on them. For example, it's intuitive that tables in Figure 1 can be classified as "education" domain, and all of the columns center around the TPE "student". Combining both explicit and implicit elements, people achieve an understanding of table context, which becomes the source of lexicon diversity in column descriptions.

2.2 REPLACE (RPL) Principles

Given a target column, the goal of RPL is to seek an alternative column name that makes sense to humans but misleads unrobust models. Gold SQL, as illustrated in Figure 1, should be correspondingly adapted by mapping the original column to its new name. In light of this, RPL should fulfill the following two principles:

ADVETA Statistics	Spider			WTQ			WikiSQL		
	Orig.	*RPL*	*ADD*	*Orig.*	*RPL*	*ADD*	*Orig.*	*RPL*	*ADD*
Basic Statistics									
#Total tables	81	81	81	327	327	327	2,716	2,716	2,716
#Avg. columns per table	5.45	–	–	6.31	–	–	6.41	–	–
#Avg. perturbed columns per table	–	2.62	3.64	–	2.65	3.27	–	3.70	4.44
#Avg. candidates per column	–	3.33	3.97	–	2.90	3.55	–	3.32	3.97
#Unique columns	211	911	1,061	527	1,656	2,976	2,414	10,787	10,474
#Unique vocab	199	598	782	596	1,156	1,459	2,414	4,147	5,099
Analytical Statistics									
#Unique semantic meanings	144	144	**683**	156*	156*	**702***	203*	203*	818*
#Avg. col name per semantic meaning	1.35	**6.33**	1.55	1.59*	**5.87***	1.64*	1.67*	**6.12***	1.52*

Table 1: ADVETA statistics comparison between original and RPL/ADD-perturbed dev set. The * mark denotes that results are based on at most 100 randomly sampled tables and obtained by manual count.

Semantic Equivalency: Under the table context of target column, substituted column names are expected to convey equivalent semantic meaning as the original name.

Phraseology Correctness: ATP aims to be natural and realistic and does not target worst-case attacks. Therefore, replaced column names are expected to follow linguistic phraseology conventions: *(i)* Grammar Correctness: Substituted column names should be free from grammar errors. *(ii)* Proper Collocation with TPE: New column names should collocate properly with TPE. For example, *height* and *tallness* both collocate well with student (TPE), but conventionally not *altitude*. *(iii)* Idiomaticity: New column names should sound natural to a native speaker to address target columns. For example, *runner-up* means *second place*, and *racer-up* is a bad replacement despite *runner* is synonymous to *racer*.

2.3 ADD Principles

ADD perturbs tables with introductions of new columns. Instead of adding random columns that fit well into the table domain, we pertinently add adversarial columns with respect to a target column for the sake of adversarial efficiency. Gold SQL should remain unchanged after ADD perturbations [1]. Below states ADD principles:

Semantic-association & Domain-relevancy: Given a target column and its table context, newly added columns are expected to *(i)* fit nicely into the table context; *(ii)* have high semantic associations with the target column yet low semantic equivalency (e.g. *sales vs. profits*, *editor vs. author*).

Phraseology Correctness: Same as RPL, columns should obey human language conventions.

Irreplaceability: Unlike RPL, any added columns should be irreplaceable with any original table columns. In other words, ADD requires semantic equivalency to be filtered out from highly semantic associations. Otherwise, the original gold SQL will not be the single correct output, which makes the perturbation unreasonable.

3 ADVETA Benchmark

Following RPL and ADD principles, we manually curate the **ADVE**rsarial **T**able perturb**A**tion (**ADVETA**) benchmark based on three mainstream Text-to-SQL datasets, Spider (Yu et al., 2018), WikiSQL (Zhong et al., 2017) and WTQ (Papernot et al., 2017). For each table from the original *development set*, we conduct RPL/ADD annotation separately, perturbing only table columns. For its associated NL-SQL pairs, we leave the NL questions unchanged and adapt gold SQLs accordingly. As a result, ADVETA consists of 3 (Spider/WTQ/WikiSQL) $*$ 2 (RPL/ADD) = 6 subsets. We next introduce annotation details and characteristics of ADVETA.

3.1 Annotation Steps

Five vendors join the annotation process. Each base dev set is split into small chunks and is manually annotated by one vendor and reviewed by another, with an inter-annotator agreement to resolve annotation inconsistency.

Before annotation, vendors are first trained to understand table context as described in § 2, then are further instructed of the following details.

RPL: RPL principles are the mandatory requirements. During annotation, vendors are given full Google access to ease the conception of synonymous names for a target column. **ADD**: ADD principles will be the primary guideline. Unlike free-style RPL annotations, vendors are provided with

[1] We omit cell value alignment in ADD for simplicity.

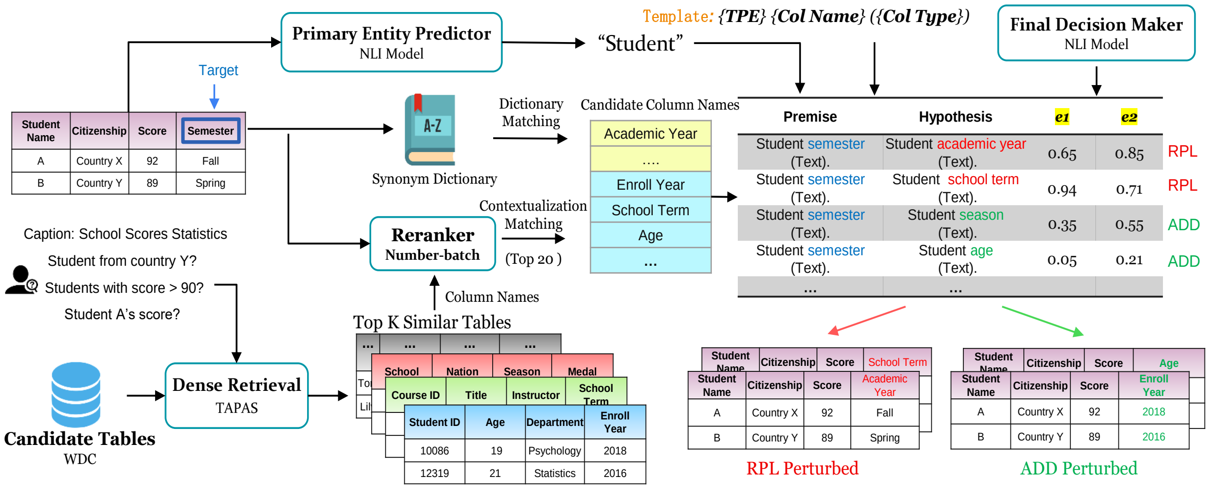


Figure 2: Overview of our CTA framework. In rare cases where TPE is missing, we apply *Primary Entity Predictor* (addressed in B.2). Otherwise we simply use annotated TPE. e_1 is obtained with premise-hypothesis as input; e_2 with hypothesis-premise.

a list of 20 candidate columns from where they select 3-5 based on semantic-association[2] Notice that we only consider columns mentioned at least once across NL questions to avoid vain efforts. In Appendix A, We display some representative benchmark annotation cases.

3.2 ADVETA Statistics and Analysis

We present comprehensive benchmark statistics and analysis results in Table 1. Notice that we limit the scope of statistics only to perturbed columns (as marked by *#Avg. perturbed col per table*).

Basic Statistics reflects elementary information of ADVETA. ***Analytical Statistics*** illustrate highlighted features of ADVETA compared with original dev-sets: *(i)* Diverse column names for a single semantic meaning: each table from the RPL subset contains approximately five times more lexicons which are used to express a single semantic meaning[3]. *(ii)* Table concept richness: each table from ADD subset contains roughly five times more columns with unique semantic meanings.

4 Contextualized Table Augmentation

In this section, we introduce our **C**ontextualized **T**able **A**ugmentation (CTA) framework as an adversarial training example generation approach tailored for tabular data. The philosophy of adversarial example generation is straightforward: Pushing augmented RPL/ADD lexicon distributions closer to human-agreeable RPL/ADD distributions. This requires maximization of lexicon diversity under the constraints of domain relevancy and clear differentiation between semantic association & semantic equivalency, as stated in ADD principle from § 2.

Well-established text adversarial example generation approaches, such as TextFooler (Jin et al., 2020) and BertAttack (Li et al., 2020), might fail to meet this objective because: *(i)* They rely on syntactic information (e.g. POS-tag, dependency, semantic role) to perform text transformations. However, such information is not available in structured tabular data, leading to poor-quality adversarial examples generated by these approaches. *(ii)* They perform sequential word-by-word transformations, which could narrow lexicon diversity (e.g. *written by* will not be replaced by *author*). *(iii)* They cannot leverage tabular context to ensure domain relevancy. *(iv)* They generally fail to distinguish semantic equivalency from high semantic association according to our observations (e.g., fail to distinguish *sales vs. profits*).

To tackle these challenges, we construct the CTA framework. Given a **target column** from a table with NL questions, *(i)* a **dense table retriever** properly contextualizes the input table, thereby pinpointing top-k most *domain-related* tables (and columns) from a large-scale database while *boosting lexicon diversity*. *(ii)* A **reranker** further narrows down *semantic-association* and produces coarse-grained ADD/RPL candidates. *(iii)* **NLI decision maker** *distinguishes semantic equivalency from semantic association* and allocates candidate

[2] We generate the candidate list with a retriever-reranker combo from § 4.

[3] For example, column names {*Last name, Family name, Surname*} express a single semantic meaning. In practice, we random sample at most 100 tables from each split, and obtain the number of unique semantic meanings by manual count.

columns to RPL/ADD buckets. A detailed illustration of our CTA framework is shown in Figure 2. We next introduce each component of CTA.

4.1 Dense Retrieval for Similar Tables

The entire framework starts with a dense retrieval module to gather most domain-related tables of user queries. We utilize the Tapas-based (Herzig et al., 2020) dense retriever in this module (Herzig et al., 2021), due to its better tabular contextualization expressiveness over classical retrieval methods such as Word2Vec (Mikolov et al., 2013) and BM25 (Robertson, 2009). Following the original usage proposed by Herzig et al. (2020), we retrieve the top 100 most domain-related tables from the backend Web Data Commons (WDC) (Lehmberg et al., 2016) database consisting of 600k non-repetitive tables with at most $five$ columns.

4.2 Numberbatch Reranker

From these retrieved domain-related tables, we further narrow down the range of most semantically associated candidate columns. This is done by a ConceptNet Numberbatch word embedding (Speer et al., 2017) reranker, who computes the cosine similarity score for a given column pair. We choose ConceptNet Numberbatch due to its advantage of far richer (520k) in-vocabulary multi-grams compared with Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and Counter-fitting (Mrkšić et al., 2016), which is especially desirable for multi-gram columns. We keep the top 20 similar among them as RPL/ADD candidates for each column of the original table.

4.3 Word-level Replacement via Dictionary

Aside from candidates obtained from retriever-reranker for whole-column level RPL, we consider word-level RPL for a target column as a complement. Specifically, we replace each word in a given target column with its synonyms recorded in the Oxford Dictionary (noise is more controllable compared with synonyms gathered by embedding). With a probability 25% for each original word to remain unchanged, we sample until the max predefined number (20) of candidates is reached or 5 consecutively repeated candidates are produced.

4.4 NLI as Final Decision Maker

So far we have pinpointed candidate columns whose domain relevancy and semantic association are already guaranteed. The final stage is to determine which one of RPL/ADD candidates is more suitable for based on its semantic equivalent against target column. Therefore, we leverage RoBERTa-MNLI (Liu et al., 2019; Williams et al., 2017), the expert in *differentiating semantic equivalency from semantic association*[4]. Practically, we construct premise-hypothesis by contextualized columns and judge semantic equivalency based on output bidirectional entailment scores e_1 and e_2.

NLI Premise-Hypothesis Construction The Quality of premise-hypothesis plays a key factor for NLI's functioning. We identify three potentially useful elements for contextualizing columns with surrounding table context: TPE, column type, and column cell value. Through manual experiments, we observe that: *(i)* Adding cell value significantly hurt decision accuracy of NLI models. *(ii)* TPE is the most important context information and cannot be ablated. *(iii)* Column type information can be a desirable source for word-sense disambiguation. Thus the final template for premise-hypothesis construction as python formatted string is expressed as: **f"{TPE} {CN} ({CT}).**", where **CN** is column name, and **CT** is column type.

RPL/ADD Decision Criterion In practice, we observe a discrepancy in output entailment scores between *premise-hypothesis* score e_1 and *hypothesis-premise* score e_2. Thus we take scores from both direction into consideration. For RPL, we empirically choose $min(e_1, e_2) >= 0.65$ (Figure 2) as the final RPL acceptance criterion to reduce occurrences of false positive entailment decision. For ADD, the criterion is instead $max(e_1, e_2) <= 0.45$ to reduce false negative entailment decisions[5].

5 Experiments and Analysis

5.1 Experimental Setup

Datasets and Models The five original Text-to-SQL datasets involves in our experiments are: Spider (Yu et al., 2018), WikiSQL (Zhong et al., 2017), WTQ (Shi et al., 2020)[6], CoSQL (Yu et al., 2019a) and SParC (Yu et al., 2019b). Their corresponding perturbed tables are from our ADVETA

[4] We ***highly recommend*** reading our pilot study in B.1.

[5] To avoid semantic conflict between a new column $\tilde{c}$ and original columns $c_1, \cdots, c_n$, we apply to each pair of $(\tilde{c}, c_i)$.

[6] Note that we use the version with SQL annotations provided by Shi et al. (2020) here, since the original WTQ (Pasupat and Liang, 2015) only contains answer annotations.

Dataset	Baseline	Dev	RPL	ADD
Spider	DuoRAT	69.9	23.8 ± 2.1 (-46.1)	36.4 ± 1.3 (-33.5)
	ETA	70.8	27.6 ± 1.8 (-43.2)	39.9 ± 0.9 (-30.9)
WikiSQL	SQLova	81.6	27.2 ± 1.3 (-54.4)	66.2 ± 2.3 (-15.4)
	CESQL	84.3	52.2 ± 0.9 (-32.1)	71.2 ± 1.5 (-13.1)
WTQ	SQUALL	44.1	22.8 ± 0.5 (-21.3)	32.9 ± 0.8 (-11.2)
CoSQL	EditSQL	39.9	13.3 ± 0.7 (-26.6)	30.5 ± 1.1 (-9.4)
	IGSQL	44.1	16.4 ± 1.2 (-27.7)	32.8 ± 2.1 (-11.3)
SParC	EditSQL	47.2	30.5 ± 0.9 (-16.7)	40.2 ± 1.2 (-7.0)
	IGSQL	50.7	34.2 ± 0.5 (-16.5)	42.9 ± 1.7 (-7.8)

Table 2: Results on original dev and ADVETA (RPL and ADD subsets). Red fonts denote *absolute percentage* performance drop compared with original dev.

benchmark. WikiSQL and WTQ are single-table, while Spider, CoSQL, and SParC have multi-tables. CoSQL and SParC are known as multi-turn Text-to-SQL datasets, sharing the same tables with Spider. Dataset statistics are shown in Appendix Table 11.

We evaluate open-source Text-to-SQL models that reach competitive performance on the aforementioned datasets. DuoRAT (Scholak et al., 2021) and ETA (Liu et al., 2021) are baselines for Spider; SQUALL (Shi et al., 2020) is the baseline for WTQ; SQLova (Hwang et al., 2019) and CESQL (Guo and Gao, 2019) are baselines for WikiSQL. For the two multi-turn datasets (CoSQL & SParC), baselines are EditSQL (Zhang et al., 2019) and IGSQL (Cai and Wan, 2020). *Exact Match* (**EM**) is employed for evaluation metric across all settings. Training details are shown in C.2.

5.2 Attack

Attack Details All baseline models are trained from scratch on corresponding original training sets, and then independently evaluated on original dev sets, ADVETA-RPL and ADVETA-ADD. Since columns have around *three* manual candidates in ADVETA-RPL/ADD, the number of possible perturbed tables scales exponentially with column numbers for a given table from the original dev set. Therefore, models are evaluated on ADVETA-RPL/ADD by sampling perturbed tables. For ***each*** NL-SQL pair and associated table(s), we sample ***one*** RPL-perturbed table and ***one*** ADD-perturbed table in each attack. **Each** column mentioned from gold SQL is perturbed by a randomly sampled manual candidate from ADVETA. For performance stability and statistical significance, we run five attacks with random seeds for each NL-SQL pair.

Attack Results Table 2 presents the performance of models on original dev sets, ADVETA-RPL and ADVETA-ADD. Across various task formats, domains, and model designs, state-of-the-art Text-to-SQL parsers experience dramatic performance drop on our benchmark: by RPL perturbations, the relative percentage drop is as high as 53.1%, whereas on ADD the drop is 25.6% on average[7]. Another interesting observation is that RPL consistently leads to higher performance drops than ADD. This is perhaps due to models' heavy reliance on lexical matching, instead of true understanding of language and table context. Conclusively, Text-to-SQL models are still far less robust than desired against variability from the table input side.

Attack Analysis To understand the reasons for parsers' vulnerability, we specifically analyze their schema linking modules which are responsible for recognizing table elements mentioned in NL questions. This module is considered a vital component for Text-to-SQL (Wang et al., 2020; Scholak et al., 2021; Liu et al., 2021). We leverage the oracle schema linking annotations on Spider (Lei et al., 2020) and test ETA model on ADVETA using the oracle linkings. Note that we update the oracle linkings accordingly when testing on RPL. Table 4 compares the performance of ETA with or without the oracle linkings, from which we make two observations: *(i)* When guided with the oracle linkings, ETA performs much better on both RPL (27.6% → 55.7%) and ADD (39.9% → 71.3%). Therefore, the failure in schema linking is one of the essential causes for the vulnerability of Text-to-SQL parsers. *(ii)* Even with the oracle linkings, the performance of ETA on RPL and ADD still lags behind its performance on the original dev set, especially on RPL. Through a careful analysis on failure cases, we find that ETA still generates table elements that have a high degree of lexical matching with NL questions, even though the correct table elements are specified in the oracle linkings.

5.3 Defense

Defense Details We carry defense experiments with SQLova, SQUALL and ETA on WikiSQL, WTQ and Spider, respectively. We compare CTA

[7] Average relative performance presented in Appendix C.3.

Approach	WikiSQL			WTQ			Spider		
	Dev	RPL	ADD	Dev	RPL	ADD	Dev	RPL	ADD
Orig.	81.6	27.2±1.3	66.2±2.3	44.1	22.8±0.5	32.9±0.8	70.8	27.6±1.8	39.9±0.9
BA	80.1±0.2	56.8±0.8	77.9±0.5	43.9±0.3	33.6±0.4	42.8±0.7	68.1±0.5	26.9±1.1	43.1±0.7
TF	80.5±0.3	57.7±0.7	77.7±0.4	43.7±0.4	35.2±0.5	42.6±0.6	67.9±0.6	28.4±1.2	42.2±0.6
W2V	80.8±0.1	60.7±1.1	78.2±0.6	43.4±0.1	36.8±0.6	42.2±0.9	68.3±0.2	30.1±1.3	43.3±1.4
MAS	–	–	–	–	–	–	69.1±0.3	27.3±0.7	35.3±0.5
CTA	**81.2 ± 0.1**	**69.2 ± 0.5**	**79.9 ± 0.3**	**44.1 ± 0.1**	**41.8 ± 0.3**	**44.6 ± 0.5**	**69.8 ± 0.1**	**35.8 ± 0.5**	**50.6 ± 0.1**
w/o Retriver	81.0±0.2	68.1±0.2	78.1±0.5	44.0±0.2	40.6±0.2	42.1±0.3	69.7±0.3	34.7±0.5	43.0±0.8
w/o MNLI	80.6±0.3	61.3±0.5	78.6±0.2	43.8±0.1	36.9±0.3	43.1±0.2	69.6±0.2	29.8±0.2	47.8±0.2

Table 3: Defense results on ADVETA (RPL and ADD subsets). Avg. EM and fluctuations of 5 runs are reported. Orig. denotes performance without defense from Table 2.

Schema Linking	**Dev**	**RPL**	**ADD**
w/o oracle	70.8	27.6 (-43.2)	39.9 (-30.9)
w/ oracle	75.2	55.7 (-19.5)	71.3 (-3.9)

Table 4: Schema linking analysis of ETA on Spider.

with three baseline adversarial training approaches: Word2Vec (W2V), TextFooler (TF) (Jin et al., 2020), and BERT-Attack (BA) (Li et al., 2020) (details found in D.). Models are trained from scratch on corresponding *augmented* training sets. Specifically, for ***each*** NL-SQL pair, we keep the original table while generating ***one*** RPL and ***one*** ADD adversarial example. As a result, augmented training data is three times as large in the sense that each NL-SQL pair is now trained against three tables: original, RPL-perturbed, and ADD-perturbed. In addition to the adversarial training defense paradigm, we also include the manual version of Multi-Annotation Selection (MAS) by Gan et al. (2021) on Spider, using their released data. The rest evaluation process is same as attack.

Defense Results Table 3 presents model performance through various defense approaches. We get two observations: *(i) CTA consistently brings better robustness.* Compared with other approaches, CTA-augmented models have the best performance across all ADVETA-RPL/ADD settings, as well as on all original dev sets. These results demonstrate CTA can effectively improve the robustness of models against RPL and ADD perturbations while introducing fewer noises into original training sets. Interestingly, we observe that textual adversarial example generation approaches (BA, TF) are outperformed by the simple W2V approach. This verifies our analysis stated in § 4. We include a case study in Appendix B.3 on characteristics of various baselines.

Method	**Col$_P$**	**Col$_R$**	**Col$_F$**	**Tab$_P$**	**Tab$_R$**	**Tab$_F$**
ETA	85.4	36.8	51.4	61.3	63.4	62.3
W2V$_{RPL}$	86.1	40.2	54.8	70.4	72.6	71.5
CTA$_{RPL}$	**88.1**	**50.8**	**64.4**	**80.1**	**85.4**	**82.7**
ETA	86.3	60.2	70.9	71.2	75.8	73.4
W2V$_{ADD}$	86.5	63.7	73.4	75.9	82.1	78.9
CTA$_{ADD}$	**88.1**	**70.2**	**78.2**	**83.6**	**89.5**	**86.4**

Table 5: The schema linking analysis of attacking with ETA and two defense approaches, namely W2V and CTA on Spider; Col as column and Tab as table. P, R, F is short for precision, recall and F1 score, respectively.

(ii) CTA fails to bring models back to their original dev performance. Even if trained with high-quality data augmented by CTA, models could still be far worse than their original performance. This gap is highly subjected to the similarity of lexicon distribution between train and dev set. Concretely, on WikiSQL and WTQ where train and dev set have a similar domain, both RPL performance and ADD performance are brought back closer to original dev performance when augmented with CTA. On the contrary, on Spider where train-dev domains overlap less, there is still a notable gap between performance after adversarial training and the original dev performance. In conclusion, more effective defense paradigms are yet to be investigated.

Defense Analysis Following attack analysis, we conduct schema linking analysis with ETA model augmented with top 2 approaches (i.e. W2V & CTA) on Spider. We follow metric calculation of (Liu et al., 2021) and details are shown in § C.4. As shown in Table 5, both approaches improve the schema linking F_1. Specifically, CTA improves column F_1 by $3\% \sim 8\%$, and table F_1 by $13\% \sim 20\%$, compared with vanilla ETA. This reveals that improvement of robustness can be primarily attributed to better schema linking.

Some might worry about the validity of the CTA's effectiveness due to data leakage risks in-

Model	Spider	Spider-Syn
RAT-SQL$_{\text{BERT}}$ (Wang et al., 2020)	69.7	48.2
RAT-SQL$_{\text{BERT}}$+MAS (Gan et al., 2021)	67.4	62.6
ETA (Liu et al., 2021)	70.8	50.6
ETA+CTA	69.8	60.4

Table 6: EM on Spider/Spider-Syn dev-sets.

curred by the annotation design that vendors are given CTA-retrieved candidate list for ADD annotations. However, we emphasize that: *(i)* RPL have *NO* vulnerability to data leakage since it is entirely independent of CTA. *(ii)* The leakage risk in ADD is negligible. On the one hand, our vast-size (600k tables) backend DB supplies tremendous data diversity, maximally reducing multiple retrievals of a single table; On the other hand, CTA's superior performance on Spider, the representative feature of which is cross-domain & cross-database across train-test splits (thus makes performance gain from data leakage hardly possible), further testifies its authentic effectiveness.

5.4 CTA Ablation Study

We carry out an ablation study to understand the roles of two core components of CTA: dense retriever and RoBERTa-MNLI. Results are shown in Table 3.

CTA w/o Retriever RPL candidates are generated merely from the dictionary; ADD generation is the same as W2V baseline. Compared with complete CTA, models augmented with this setting experience $1.1\% \sim 1.2\%$ and $1.8\% \sim 7.6\%$ performance drop on RPL and ADD, respectively. We attribute RPL drops to loss of real-world lexicon diversity and ADD drops to loss of domain relevancy.

CTA w/o MNLI RPL and ADD candidates are generated in the same way as CTA, but without denoising of MNLI. RPL/ADD decisions solely rely on ranked semantic similarity. Compared with complete CTA, models augmented by this setting experience significant performance drops ($4.9\% \sim 7.9\%$) on all RPL subsets, and moderate drops ($1.5\% \sim 2.8\%$) on all ADD subsets. We attribute these drops to the inaccurate differentiation between semantic equivalency and semantic association due to lack of MNLI, which results in noisy RPL/ADD adversarial examples.

5.5 Generalization to NL Perturbations

Beyond CTA's effectiveness against table-side perturbations, a natural question follows: could retraining with adversarial table examples improve model robustness against perturbations from the other side of Text-to-SQL input (i.e., NL questions)? To explore this, we directly evaluate ETA (trained with CTA-augmented Spider train-set) on Spider-Syn dataset (Gan et al., 2021), which replaces schema-related tokens in NL question with its synonym. We observe an encouraging 9.8% EM improvement compared with vanilla ETA (trained with Spider train-set). This verifies *CTA's generalizability to NL-side perturbations*, with comparable effectiveness as the previous SOTA defense approach MAS, which fails to generalize to table-side perturbations on ADVETA in Table 3.

6 Related Work

Robustness of Text-to-SQL As discussed in § 1, previous works (Gan et al., 2021; Zeng et al., 2020; Deng et al., 2021) exclusively study robustness of Text-to-SQL parsers against perturbations in NL question inputs. Our work instead focuses on variability from the table input side and reveals parsers' vulnerability to table perturbations.

Adversarial Example Generation Existing works on adversarial text example generations can be classified into three categories: (1) Sentence-Level. This line of work generates adversarial examples by introducing distracting sentences or paraphrasing sentences (Jia and Liang, 2017; Iyyer et al., 2018). (2) Word-Level. This dimension of work generates adversarial examples by flipping words in a sentence, replacing words with their synonyms, and deleting random words (Li et al., 2020; Ren et al., 2019; Jin et al., 2020). (3) Char-Level. This line of work flips, deletes, and inserts random chars in a word to generate adversarial examples (Belinkov and Bisk, 2018; Gao et al., 2018). All the three categories of approaches have been widely used to reveal vulnerabilities of high-performance neural models on various tasks, including text classification (Ren et al., 2019; Morris et al., 2020), natural language inference (Li et al., 2020) and question answering (Ribeiro et al., 2018). Previous work on robustness of Text-to-SQL and semantic parsing models primarily adopt word-level perturbations to generate adversarial examples (Huang et al., 2021). For example, the Spider-Sync adversarial benchmark (Gan et al., 2021) is curated by replacing schema-related words in questions with their synonyms.

Despite these methods' effectiveness in generating adversarial text examples, they are not readily applicable for structural tabular data, as we discussed in § 4. Apart from this, table-side perturbations enjoy ***much higher attacking efficiency***: the attack coverage of a single table modification includes all affiliated SQLs, whereas one NL-side perturbation only affects a single SQL. Combined with the lighter cognitive efforts of tabular context understanding than NL-understanding, ATP is ***arguably lower in annotation costs***.

Previous work on table perturbations (Cartella et al., 2021; Ballet et al., 2019) focuses on table cell values; another work, (Ma and Wang, 2020) study impacts of naively (i.e., without consideration of table context information and without human guarantee) renaming irrelevant columns and adding irrelevant columns. In this work, we focus on table columns and propose an effective CTA framework that better leverages tabular context information for adversarial example generation, as well as manually annotate ADVETA benchmark.

7 Conclusion

We introduce **A**dversarial **T**able **P**erturbation (**ATP**), a new paradigm for evaluating model robustness on Text-to-SQL and define its conduction principles. We curate the ADVETA benchmark, on which all state-of-the-art models experience dramatic performance drop. For defense purposes, we design the CTA framework tailored for tabular adversarial training example generation. While CTA outperforms all baseline methods in robustness enhancement, there is still an unfilled gap from the original performance. This calls for future exploration of the robustness of Text-to-SQL parsers against ATP.

Ethical Considerations

Our ADVETA benchmark presented in this work is a free and open resource for the community to study the robustness of Text-to-SQL models. We collected tables from three mainstream Text-to-SQL datasets, Spider (Yu et al., 2018), WikiSQL (Zhong et al., 2017) and WTQ (Papernot et al., 2017), which are also free and open datasets for research use. For the table perturbation step, we hire professional annotators to find suitable RPL/ADD candidates for target columns. We pay the annotators at a price of 10 dollars per hour. The total time cost for annotating our benchmark is 253 hours.

All the experiments in this paper can be run on a single Tesla V100 GPU. Our benchmark will be released along with the paper.

References

Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal*, 28:793 – 819.

Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. 2019. Imperceptible adversarial attacks on tabular data. *CoRR*, abs/1911.03274.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. *ArXiv*, abs/1711.02173.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326.

Yitao Cai and Xiaojun Wan. 2020. IGSQL: Database schema interaction graph based neural model for context-dependent text-to-SQL generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6903–6912, Online. Association for Computational Linguistics.

Francesco Cartella, Orlando Anunciação, Yuki Funabiki, Daisuke Yamaguchi, Toru Akishita, and Olivier Elshocht. 2021. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. In *Proceedings of the Workshop on Artificial Intelligence Safety 2021 (SafeAI 2021) co-located with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, February 8, 2021*, volume 2808 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. *CoRR*, abs/2105.07464.

Yujian Gan, Xinyun Chen, Qiuping Huang, Matthew Purver, John R. Woodward, Jinxia Xie, and Pengsheng Huang. 2021. Towards robustness of text-to-SQL models against synonym substitution. 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 2505–2515, Online. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821.

Tong Guo and Huilin Gao. 2019. Content enhanced bert-based text-to-sql generation. *arXiv preprint arXiv:1910.07179*.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. TAPAS: weakly supervised table parsing via pre-training. *CoRR*, abs/2004.02349.

Felix Hill, Kyunghyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. 2015a. Embedding word similarity with neural machine translation.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015b. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Shuo Huang, Zhuang Li, Lizhen Qu, and Lei Pan. 2021. On robustness of neural semantic parsers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3333–3342, Online. Association for Computational Linguistics.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. 2019. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 75–76, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the role of schema linking in text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954, Online. Association for Computational Linguistics.

Fei Li and H. V. Jagadish. 2016. Understanding natural language queries over relational databases. *SIGMOD Record*, 45:6–13.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Qian Liu, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, and Jian-Guang Lou. 2021. Awakening latent grounding from pretrained language models for semantic parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1174–1189, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Pingchuan Ma and Shuai Wang. 2020. Mt-teql: Evaluating and augmenting consistency of text-to-sql models with metamorphic testing.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. *CoRR*, abs/1603.00892.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

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)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

S. Robertson. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Torsten Scholak, Raymond Li, Dzmitry Bahdanau, Harm de Vries, and Chris Pal. 2021. DuoRAT: Towards simpler text-to-SQL models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1313–1321, Online. Association for Computational Linguistics.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1849–1864, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. pages 4444–4451.

Sai Sumathi and S Esakkirajan. 2007. *Fundamentals of relational database management systems*, volume 47. Springer.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *CoRR*, abs/1909.00161.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019a. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019b. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

Jichuan Zeng, Xi Victoria Lin, Caiming Xiong, Richard Socher, Michael R Lyu, Irwin King, and Steven CH Hoi. 2020. Photon: A robust cross-domain text-to-sql system. *arXiv preprint arXiv:2007.15280.*

Rui Zhang, Tao Yu, Heyang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. Editing-based SQL query generation for cross-domain context-dependent questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5338–5349, Hong Kong, China. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103.*

A Benchmark Examples

We display some representative benchmark annotation cases for to convey readers a intuitive feeling on our RPL and ADD subsets. As reflected in Figure 3, RPL reflects the following characteristics beyond RPL principles: *(i)* Abbreviation of common words. e.g. *Cell number* vs. *Tel.* *(ii)* Idiomatic transformation e.g. *Air date* vs. *Release time (iii)* Part of speech structure transformation e.g. *Written by* vs. *Author*. ADD perturbations faithfully obey ADD principles and additions demonstrate high coherency with respect to original domain.

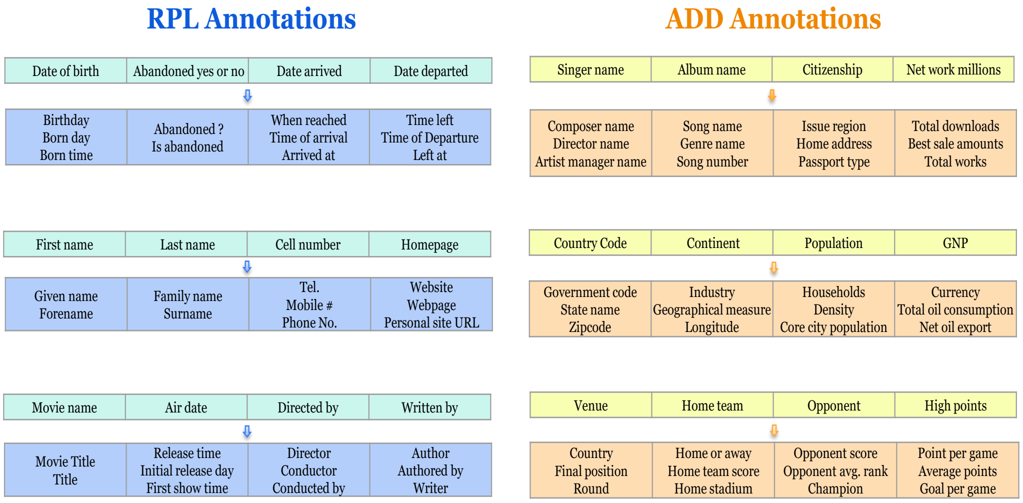
B CTA Details

B.1 NLI-based Substitutability Verification

Approach	e_1	e_2	Δ_{e_1}	Δ_{e_2}
Roberta-RTE				
human	**48.5**	48.1	0.65	0.46
embedding	45.7	45.6	0.26	0.30
ranodm	43.0	42.8	0.53	0.70
Roberta-SNLI				
human	74.5	74.1	0.48	0.61
embedding	56.7	66.0	0.75	0.37
ranodm	31.2	30.9	0.78	0.64
Roberta-MNLI				
human	**77.1**	76.4	0.86	0.36
embedding	**52.2**	58.7	0.34	0.69
ranodm	**16.5**	14.8	0.50	0.49

Table 7: Average foward entailment score e_1, backward entail e_2, and corresponding standard deviations across 9 settings. In all human annotation cases, higher entailment is better. In all random replacement cases, lower is better.

Implementation Details For each pair of target column and candidate column, we contextualize each column with the template described in Premise-Hypothesis Construction from section § 4. Then with the contextualized target column as the premise and the contextualized RPL candidate as the hypothesis, the NLI model computes both forward entailment score $e1$ and backward score $e2$. Notice that $e2$ computation takes the contextualized RPL candidate as premise and the contextualized target column as hypothesis in input. We obtain entailment scores from both directions because of the observed score fluctuation caused by reversion in practicable cases.



RPL Annotations

Date of birth	Abandoned yes or no	Date arrived	Date departed
Birthday Born day Born time	Abandoned ? Is abandoned	When reached Time of arrival Arrived at	Time left Time of Departure Left at

First name	Last name	Cell number	Homepage
Given name Forename	Family name Surname	Tel. Mobile # Phone No.	Website Webpage Personal site URL

Movie name	Air date	Directed by	Written by
Movie Title Title	Release time Initial release day First show time	Director Conductor Conducted by	Author Authored by Writer

ADD Annotations

Singer name	Album name	Citizenship	Net work millions
Composer name Director name Artist manager name	Song name Genre name Song number	Issue region Home address Passport type	Total downloads Best sale amounts Total works

Country Code	Continent	Population	GNP
Government code State name Zipcode	Industry Geographical measure Longitude	Households Density Core city population	Currency Total oil consumption Net oil export

Venue	Home team	Opponent	High points
Country Final position Round	Home or away Home team score Home stadium	Opponent score Opponent avg. rank Champion	Point per game Average points Goal per game

Figure 3: RPL and ADD annotation examples from our ATP benchmark. Rows with shallow colors are original headers, whereas those deep-shaded ones are our human annotations.

Pilot Study for Model Ability We carry out a pilot study to test NLI models' capability of differentiating semantic equivalency and similarity in this section. RoBERTa (Liu et al., 2019) is chosen as the backbone model due to its outstanding performance and computational efficiency across various NLI datasets. Fine-tuned RoBERTa on three well-known NLI datasets: RTE (Dagan et al., 2013), SNLI (Bowman et al., 2015), and MNLI (Williams et al., 2017) are compared to demonstrate model ability difference due to training data,.

We consider three levels of substitutability, from highest to lowest: human manual substitution (human-annotated replacements sampled from benchmark RPL subsets), embedding-based substitution (top-10 similar multi-grams from ConceptNet Numberbatch word embedding (Speer et al., 2017)), and random substitution (randomly sampled columns across benchmark(Speer et al., 2017)). Practically, we randomly sample 1000 pairs of data each time and repeat each setting five times.

We report the both average forward e_1 and backward entailment scores e_2, as well their standard deviations for each setting across five runs (table 8). It is immediately obvious that RoBERTa-MNLI surpasses other models in verbal dexterity: the entailment score correlates best with true degrees of substitutability.

Performance on SimLex-999 *SimLex-999* (Hill et al., 2015b) is a gold standard resource for measuring how well models capture similarity, rather than relatedness or association between a input pair of words (e.g. *cold* and *hot* are closely associated but definitely not similar). Thus it is especially suitable for our purpose of further testing RoBERTa-MNLI's ability of semantic discrimination. We treat the entailment score produced by the model as its judgment of semantic similarity and report its Pearson correlation against the ground truth similarity score. Results suggest that RoBERTa-MNLI is quite competitive at discriminating association and relatedness from similarity.

Approach	ρ
Word2Vec (Mikolov et al., 2013)	0.37
Glove (Pennington et al., 2014)	0.41
Glove + Counter-fitting (Mrksic et al., 2016)	0.58
NMT Emedding (Hill et al., 2015a)	0.58
aragram-SL999 (Wieting et al., 2015)	0.69
RoBERTa-MNLI (ours)	**0.70**

Table 8: Results on SimLex-999. ρ (Perason correlation) is used as the primary metric.

Case Study To test the hard case performance of RoBERTa-MNLI, we come up with some tricky examples as shown in Table 9. The upper half of the table presents hard ***replaceable*** cases that emphasize idiomatic transformations or word-sense disambiguation. The lower half contains hard ***irreplaceable*** cases in which phrases have a high degree of conceptual association, yet still not semantically equivalent. Results reveal the surprisingly abundant and accurate lexicon knowledge condensed in RoBERTa-MNLI.

Premise	Hypothesis	ENT	NON-ENT
Replaceable			
Runner-up.	Second place.	**97.1**	2.9
First name.	Given name.	**93.7**	6.3
Airline code.	Airline number.	**82.3**	17.7
Cartoon air date.	Cartoon release time.	**91.4**	8.6
Book author.	Book written by.	**97.8**	2.2
Irreplaceable			
Student height.	Student altitude.	26.9	**73.1**
Company sales.	Company profits.	1.9	**98.1**
People killed.	People injured.	2.1	**97.9**
Population number.	Population code.	37.1	**62.9**
Political party.	Political celebration.	27.5	**72.5**

Table 9: Hard cases we come up with to explore upper-bounds of Roberta-MNLI ability. ENT as entaiment score, NON-ENT as contradiction + neutral score. Score of expected label is bolded.

B.2 Zero-shot TPE Classification

We build the previous premise-hypothesis construction in § 4.4 based on the assumption of availability of TPE, which is frequently not true. Thus our goal is to make a reasonable prediction on TPE for those missing cases. Practically, we make use HuggingFace (Wolf et al., 2020) implementation of zero-shot text classification (Yin et al., 2019) to classify missing TPE into 48 pre-defined categories with the input of concatenated table caption, columns, and cell values.

Implementation Details Based on the 60+ fine-grained categories defined in Few-NERD (Ding et al., 2021), We modify and integrate them into 48 classes as candidate labels ($|L| = 48$). With a Roberta-MNLI as the workhorse model, our overall modeling process is modeled as

$$\tilde{c_t} = \arg\max_i \frac{\exp(f_\theta(\mathbf{L}_i \mid d; \mathbf{c}; \mathbf{v}; d)_{ent})}{\sum_{j \in |L|} \exp(f_\theta(\mathbf{L}_j \mid d; \mathbf{c}; \mathbf{v})_{ent})}$$

where $\mathbf{c}$ is column names, $\mathbf{v}$ is a randomly selected column value affiliated with a given column, and d is table captions for a given table. Roberta-MNLI (annotated as f_θ) outputs raw logits of contradiction, neutral, and entailment scores. Softmax is finally applied entailment logits across 48 categories, with the top 1 label as final the primary entity prediction.

Human evaluation We randomly sample 100 tables from our benchmark and ask three vendors to rate the reasonability of each predicted TPE from a scale of $1-5$. 1 as totally unreasonable, 3 as mildly acceptable, and 5 as perfectly parallel with human guesses. We average out the rating from all three vendors and get a result of *4.13*. This indicates the practicability of zero-shot TPE classification.

B.3 Perturbation Case Study

In this section, we present a case study on adversarial training examples generated by CTA and baseline approaches in Table 10. We can make the following observations: *(i)* CTA tend to produce less low-frequency words (e.g. *padrone, neosurrealist*) in both RPL and ADD i.e. lower perplexity. *(ii)* CTA-generated samples fit better with the specificity level of table columns. For example, RPL pair *(region, sphere)* is overly broadened, whereas names such *ballads denomination, supermanager, thespian* might be overly specified to fit into table headers. *(iii)* CTA incurs least semantic drift in RPL. In all baseline methods, there is a good chance to observe semantic-distinctive pairs such as *(region, member)*, *(type, number)*, *(type, guy)*. With CTA, such risk is minimal.

C Experimental Details

C.1 Original Datasets statistics

The detail statistics of five Text-to-SQL datasets are shown in Table 11. According to CoSQL (Yu et al., 2019a) and SParC (Yu et al., 2019b) paper, the two multi-turn Text-to-SQL datasets share the same tables with Spider (Yu et al., 2018).

C.2 Baseline Details

SQLova For all defense results of the WikiSQL dataset, we employ the SQLova model, whose official codes are released in (Hwang et al., 2019). We use uncased BERT-large as the encoder. The learning rate is 1×10^{-3} and the learning rate of BERT-large is 1×10^{-5}. The training epoch is 30 with a batch size of 12. The training process lasts 12 hours on a single 16GB Tesla V100 GPU.

SQUALL We employ the SQUALL model, following (Shi et al., 2020), to get all defense results of the WTQ dataset. The training epoch is 20 with a batch size of 30; The dropout rate is 0.2; The training process lasts 9 hours on a single 16GB Tesla V100 GPU.

ETA We implement the ETA model following (Liu et al., 2021). We use an uncased BERT-large whole word masking version as the encoder. The learning rate is 5×10^{-5} and the training epoch is 50. The batch size and gradient accumulation steps

Perturbation	Table Context	BA	TF	W2V	CTA
RPL	club id	member	districts	regionary	place
	region	regional	zones	location	location
	name	district	sphere	regions	district
	author id	types	guy	typeful	category
	type	number	genus	example	genre
	title	style	categories	sort	kind
	singer id	songs title	ballads denomination	name	music name
	song name	singer name	ballads appointments	polynymous	song title
	country	chorus name	song designation	folk-song name	music designation
ADD	course id	classes	sophomore	studential	school
	semester	honors	majoring	intersession	enrollment
	section id	session	freshman	undergraduate	university
	artist id	composition	musicianship	tachiste	publisher
	artist	creator	thespian	neosurrealist	album
	age	design	arranger	creative person	genre
	movie id	designer	officers	corporate leader	producer
	director	operator	padrone	supermanager	scenarist
	year	composer	guide	executive	writer

Table 10: Adversarial training examples generated by CTA and baseline approaches. Words with red color font are target columns.

Datasets	Train			Dev		
	#T	*#Q*	*#Avg. Col*	*#T*	*#Q*	*#Avg. Col*
WTQ	1,290	9,030	6.39	327	2,246	6.41
WikiSQL	18,590	56,355	6.40	2,716	8,421	6.31
Spider	795	6,997	5.52	81	1,034	5.45
CoSQL	795	9,478	5.52	81	1,299	5.45
SParC	795	12,011	5.52	81	1,625	5.45

Table 11: Original datasets statistics. *#T* represents total number of tables in a dataset (*#Q* for questions). *#Avg. Col* stands for avg. number of columns per table. Spider, CoSQL and SParC share the same tables.

are 6 and 4. The training process lasts 24 hours on a single 32GB Tesla V100 GPU.

C.3 Attack Performance Calculation Details

Table 12 shows the attack performance of RPL and ADD perturbations. In this section, we show the calculation details of the average attack relative performance drop. For example, on the Spider dataset, the relative performance drop of the DuoRAT model against RPL perturbation is 65.9%, and 61.0% for the ETA model. For RPL perturbation, we average out the relative performance drop of 9 models and report the average relative percentage drop (53.1%). Same as RPL, we get the average relative percentage drop (25.6%) for ADD perturbation.

Dataset	**Baseline**	**Dev**	**RPL**	**ADD**
Spider	DuoRAT	69.9	23.8 ± 2.1 (-46.1 / -65.9%)	36.4 ± 1.3 (-33.5 / -47.9%)
	ETA	70.8	27.6 ± 1.8 (-43.2 / -61.0%)	39.9 ± 0.9 (-30.9 / -43.6%)
WikiSQL	SQLova	81.6	27.2 ± 1.3 (-54.4 / -66.7%)	66.2 ± 2.3 (-15.4 / -18.9%)
	CESQL	84.3	52.2 ± 0.9 (-32.1 / -38.1%)	71.2 ± 1.5 (-13.1 / -15.5%)
WTQ	SQUALL	44.1	22.8 ± 0.5 (-21.3 / -48.3%)	32.9 ± 0.8 (-11.2 / -25.4%)
CoSQL	EditSQL	39.9	13.3 ± 0.7 (-26.6 / -66.7%)	30.5 ± 1.1 (-9.4 / -23.6%)
	IGSQL	44.1	16.4 ± 1.2 (-27.7 / -62.8%)	32.8 ± 2.1 (-11.3 / -25.6%)
SParC	EditSQL	47.2	30.5 ± 0.9 (-16.7 / -35.4%)	40.2 ± 1.2 (-7.0 / -14.8%)
	IGSQL	50.7	34.2 ± 0.5 (-16.5 / -32.5%)	42.9 ± 1.7 (-7.8 / -15.4%)

Table 12: The Exact Match Accuracy on the development set and ADVETA. Red font denotes the absolute(left) and relative(right) performance drop percentage compared with original dev accuracy.

C.4 Schema Linking Calculation

We follow the work of Liu et al. (2021) to measure the performance of ETA schema linking predictions. Let Ω_{col} be a set $\{(c, q)_i | 1 \leq i \leq N\}$ which contains N gold (column-question token) tuples. Let $\overline{\Omega}_{col}$ be a set $\{(\overline{c}, \overline{q})_j | 1 \leq j \leq M\}$ which contains M predicted (column-question token) tuples. We define the precision(Col$_P$), recall(Col$_R$), F1-score(Col_F) as:

$$\frac{|\Gamma_{col}|}{|\overline{\Omega}_{col}|}, \frac{|\Gamma_{col}|}{|\Omega_{col}|}, \frac{2\text{Col}_P\text{Col}_R}{\text{Col}_P + \text{Col}_R}$$

where $\Gamma_{col} = \Omega_{col} \bigcap \overline{\Omega}_{col}$. The definitions of Tab_P, Tab_R, Tab_F are similar.

D Baseline Approach Details

W2V To generate candidates for a given column, W2V randomly samples $five$ candidates from the top 15 cosine-similar (Numberbatch word embeddings) for RPL and 15-50 for ADD. Textfooler and BERT-Attack also follow this hyper-parameter setting. For both TextFooler and BERT-Attack, we skip their word importance ranking (WIR) modules while only keeping the word transformer modules for candidate generation[8].

TextFooler TextFooler is one of the state-of-the-art attacking frameworks for discriminative tasks on unstructured text. We skip its word importance ranking (WIR) step since our target column has already been located. Its word transformer module is faithfully re-implemented to generate candidates for a target column. Counter-fitted word embedding (Mrksic et al., 2016) are used for similarity computation, and modified sentences are constrained by both POS-tag consistency and Sim-CSE (Gao et al., 2021) similarity score.

BERT-Attack BERT-Attack is another representative text attacking framework. Similar to our adaptation of TextFooler, we skip WIR and only keep the core masked language model-based word transformation. Following original work, low-quality or sub-word tokens predicted by BERT-Large are discarded; perturbed sentence similarities compared with the original are guaranteed by Sim-CSE.

[8] We contextualize columns with templates that additionally considers cell values and POS-tag consistency.