On Robustness of Neural Semantic Parsers

Shuo Huang[1], Zhuang Li[1], Lizhen Qu[1]‡, Lei Pan[2]
Faculty of Information and Technology, Monash University[1]
School of Info Technology, Deakin University[2]
shua0043@student.monash.edu
Zhuang.Li@monash.edu
Lizhen.Qu@monash.edu
l.pan@deakin.edu.au

Abstract

Semantic parsing maps natural language (NL) utterances into logical forms (LFs), which underpins many advanced NLP problems. Semantic parsers gain performance boosts with deep neural networks, but inherit vulnerabilities against adversarial examples. In this paper, we provide the empirical study on the robustness of semantic parsers in the presence of adversarial attacks. Formally, adversaries of semantic parsing are considered to be the perturbed utterance-LF pairs, whose utterances have exactly the same meanings as the original ones. A scalable methodology is proposed to construct robustness test sets based on existing benchmark corpora. Our results answered five research questions in measuring the sate-of-the-art parsers' performance on robustness test sets, and evaluating the effect of data augmentation.

1 Introduction

Semantic parsing aims to map natural language (NL) utterances into logical forms (LFs), which can be executed on a knowledge base (KB) to yield denotations (Kamath and Das, 2018). At the core of the state-of-the-art (SOTA) semantic parsers are deep learning models, which are widely known to be vulnerable to *adversarial samples* (Zhang et al., 2020). This kind of examples is created by adding tiny perturbations to inputs but can severely deteriorate model performance. To the best of our knowledge, despite the popularity of semantic parsing (Kamath and Das, 2018), there is still no published work on studying the robustness of neural semantic parsers against adversarial examples. Therefore, we conduct the *first* empirical study to evaluate the effect of adversarial examples on SOTA neural semantic parsers.

‡corresponding author

Unlike other disciplines, it is unclear what adversaries are for semantic parsers. For computer vision systems, adversaries are often generated by modifying inputs with *imperceptible* perturbations. In contrast, a flip of single word or a character in an utterance can significantly change its meaning so that the changes are perceptible by humans. To address this issue, (Michel et al., 2019) argue that adversaries for sequence-to-sequence models should maximally retain meanings after perturbing inputs. However, any meaning-changing utterances are supposed to have different meaning representations. A robust semantic parser should be invariant to meaning-preserving modifications. In light of this, given a semantic parser, we define its adversaries as the perturbed utterances satisfying two conditions: i) they have *exactly* the same meanings as the original ones according to human judgements; and ii) the parser consistently produces incorrect LFs on them.

Although new evaluation frameworks are proposed for NLP tasks (Xu et al., 2020; Michel et al., 2019), there is no framework designed for assessing robustness of semantic parsers against meaning-preserving adversaries. The current evaluation metrics focus only on standard accuracy, which measures to what degree predictions match gold standards. As pointed out by (Tsipras et al., 2018), it is challenging to achieve both high standard accuracy and high robust accuracy, which measures the accuracy on adversarially perturbed examples. In order to facilitate devising novel methods for robust semantic parsers, it is desirable to develop a semantic parsing evaluation framework considering both measures.

In this work, we propose an evaluation framework for robust semantic parsing. The framework consists of an evaluation corpus and a set of customized metrics. We construct the evaluation corpus by extending three existing semantic parsing

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 3333–3342
April 19 - 23, 2021.

benchmark corpora. In order to generate meaning-preserving examples, we apply automatic methods to modify the utterances in those benchmark corpora by paraphrasing and injecting grammatical errors. Among the perturbed examples generated from the test sets, we build meaning-preserving test sets by filtering out the meaning-changing ones using crowdsourcing. The robustness of the semantic parsers is measured by a set of custom metrics, with and without adversarial training methods.

We conduct the *first* empirical study on the robustness of semantic parsing by evaluating three SOTA neural semantic parsers using the proposed framework. The key findings from our experiments are three-folds:

- None of those SOTA semantic parsers can consistently outperform the others in terms of robustness against meaning-preserving adversarial examples;
- Those neural semantic parsers are more robust to word-level perturbations than sentence-level ones;
- Adversarial training through data augmentation indeed significantly improve the robustness accuracy but can only slightly influence standard accuracy.

The generated corpus and source code are available at https://github.com/shuo956/On-Robustness-of-nerual-smentic-parsing.git

2 Related Work

Semantic Parsing The SOTA neural semantic parsers formulate this task as a machine translation problem. They extend SEQ2SEQ with attention (Luong et al., 2015) to map NL utterances into LFs in target languages (e.g., lambda calculus, SQL, Python, etc.). One type of such parsers directly generates sequences of predicates as LFs (Dong and Lapata, 2016, 2018; Huang et al., 2018). The other type of parsers utilizes grammar rules to constrain the search space of LFs during decoding (Yin and Neubig, 2018; Chen et al., 2018; Guo et al., 2019b; Wang et al., 2020). However, neither of the two types of parsers are evaluated against adversarial examples.

Adversarial Examples The adversarial examples are firstly defined and investigated in computer vision. Adversarial examples in that field are generated by adding imperceptible noise to input images, which lead to false predictions of machine learning models (Madry et al., 2018; Goodfellow et al., 2014). However, it is non-trivial to add such noise to text in natural language processing (NLP) tasks due to the discrete nature of languages. Minor changes in characters or words may be perceptible to humans and may lead to change of meanings. To date, it is still difficult to reach an agreement on the definition of adversarial examples across tasks.

Jia and Liang (2017); Belinkov and Bisk (2017); Ebrahimi et al. (2018); Miyato et al. (2016) add distracting sentences and sequences of random words into text or flip randomly characters or words in input text, which can confuse the models but do not affect the labels judged by humans. In those works, such perturbations are not required to keep the semantics of original text. (Michel et al., 2019) argue that perturbations for SEQ2SEQ tasks should minimize the change of semantic in input text but dramatically alter the meaning of outputs. (Cheng et al., 2020) uses a sentiment classifier to verify whether sentiments of the original utterances is preserved after perturbation while we use crowdsourcing to ensure the meaning remains. Adversarial examples in semantic parsing cannot simply borrow from prior work because a parser does not make errors if it generates a different and correct LF when there is any subtle change in input text leading to change of semantics. In contrast, adversarial examples are supposed to cause parsing errors. Thus, adversarial examples w.r.t. meaning-changing perturbations are not well defined.

There are two types of methods in generating adversarial examples. The white-box methods (Papernot et al., 2016; Ebrahimi et al., 2017; Athalye and Carlini, 2018) assume that the attacker have direct access to model details including their parameters, but the black-box methods assume that attackers have no access to model details except feeding input data and getting outputs from models (Gao et al., 2018; Guo et al., 2019a; Chan et al., 2018; Blohm et al., 2018).

Adversarial Training Adversarial training aims at improving the robustness of machine learning models against adversarial examples (Goodfellow et al., 2014; Miyato et al., 2016; Li et al., 2018). One line of research is to augment the training data with the adversarial examples. However, Ebrahimi

et al. (2018) points out that adversarial training may cause the model oversensitive to the adversarial examples. The other approach is to increase model capacity that may improve model's robustness (Madry et al., 2018). More techniques regarding adversarial defense can be found in the recent survey (Wang et al., 2019). In semantic parsing, data augmentation methods (Jia and Liang, 2016; Guo et al., 2018) are proposed only to improve performance of models on examples without perturbation but not to improve their robustness.

3 Evaluation Framework for Robust Semantic Parsing

A robust semantic parser aims to transduce all utterances with or without meaning-preserving perturbations into correct LFs. Formally, let $\mathbf{x} = x_1, \ldots, x_{|\mathbf{x}|}$ denote a natural language utterance, and $\mathbf{y} = y_1, \ldots, y_{|\mathbf{y}|}$ be its LF, a semantic parser estimates the conditional probability (denoted by $p(\mathbf{y}|\mathbf{x})$) of an LF $\mathbf{y}$ given an input utterance $\mathbf{x}$. A robust parser's predictions are invariant to all $\mathbf{x}'$ that are generated from $\mathbf{x}$ by meaning-preserving perturbations.

For any $(\mathbf{x}, \mathbf{y})$, let $S_{\text{perturb}}(\mathbf{x}, \mathbf{y})$ denote the set of *all* meaning-preserving perturbations of $\mathbf{x}$ that is parsed to $\mathbf{y}$:

$$S_{\text{perturb}}(\mathbf{x}, \mathbf{y}) = \{(\hat{\mathbf{x}}, \mathbf{y}) : \hat{\mathbf{x}} \in B(\mathbf{x}) \wedge o(\hat{\mathbf{x}}) = \mathbf{y}\} \quad (1)$$

where $B(\mathbf{x})$ denotes the set of *all* allowed perturbations of $\mathbf{x}$, and $o(\hat{\mathbf{x}})$ is an ideal parser that maps an utterance to its LF.

A set of meaning-preserving perburbed examples $S_{\text{perturb}}(D)$ w.r.t. a corpus D is the union of all $S_{\text{perturb}}(\mathbf{x}, \mathbf{y})$ created from D.

An adversarial example w.r.t. a semantic parser is an utterance-LF pair, which is in $S_{\text{perturb}}(\mathbf{x}, \mathbf{y})$ and is parsed into an incorrect LF by that parser. A set of adversarial examples w.r.t. a $S_{\text{perturb}}(\mathbf{x}, \mathbf{y})$ and a parser $f(\mathbf{x})$ is obtained by:

$$S_{\text{adv}}(\mathbf{x}, \mathbf{y}) = \{(\hat{\mathbf{x}}, \mathbf{y}) : f(\hat{\mathbf{x}}) \neq \mathbf{y}, \forall (\hat{\mathbf{x}}, \mathbf{y}) \in S_{\text{perturb}}(\mathbf{x}, \mathbf{y})\} \quad (2)$$

Subsequently, an adversary set w.r.t. a semantic parsing corpus D is created by taking the union of all adversary sets created from each example in D.

In the following, we present an evaluation framework for robust semantic parsing, which consists of an evaluation corpus, a set of evaluation metrics, and the corresponding toolkit for evaluating any new parsers.

3.1 Construction of the Evaluation Corpus

We construct the evaluation corpus in a scalable manner by combining the existing semantic parsing benchmark corpora. Each of such corpora will be referred to as a *domain* in the whole corpus. There are a train set, a validation set, and a standard test set in each domain. We perturb the examples in each test set to build a *meaning-preserving test set* for each domain. More specifically, each example in a meaning-preserving test set is a perturbed utterance paired with its LF before perturbation. In the following, we detail each perturbation method and how we apply the crowdsourcing to remove meaning-changing ones.

3.1.1 Meaning-Preserving Perturbations

Perturbations	Meaning-preserving Examples
Insertion	in what state is the largest in population
Deletion	what state is ~~the~~ largest in population
Substitution (f)	what state is the largest among population
Substitution (nf)	what state is the most in population
Back Translation	state with the largest population
Reordering	the largest population is in what state

Table 1: The meaning-preserving examples of the original utterance "what state is the largest in population".

Given an utterance, we perturb it by performing four different word-level operations and two sentence-level operations, *respectively*. Table 1 lists the examples of generated meaning-preserving examples categorized according to the respective generation methods. More details are given below.

Insertion Given an utterance $\mathbf{x}$, we randomly select a word position $t' \in \{1, \ldots, |\mathbf{x}|\}$. A meaning-preserving example $\mathbf{x}'$ is created by inserting a function word at position t'.

Deletion We randomly remove a function word x_t from $\mathbf{x}$.

Substitution (f) Every function word in $\mathbf{x}$ is replaced with a random but different function word to generate a perturbed example.

Substitution (nf) For every non-function word in $\mathbf{x}$, we apply the pretrained language model ELECTRA (Clark et al., 2020) to select top-k candidate words and exclude the original word. We generate a perturbed utterance for each valid candidate word. Since this method may generate utterances far from their original meaning, we subsequently filter those utterances by measuring their semantic similarity with the original ones.

Specifically, we apply the SOTA sentence similarity model Sentence-Bert (Reimers and Gurevych, 2019) to compute similarity scores for each generated utterance resulting in only the n highest scored utterances.

Back Translation Inspired by (Lichtarge et al., 2019), we revise utterances using back translation. We apply the Google translation API to translate utterances into a bridge language and then translate them back to English. Russian, French, and Japanese are used as the bridge languages to diversify and maximize the coverage of meaning-preserving perturbations. We select the best translation among all three translations for each original utterance according to Sentence-Bert's scores.

Reordering Similar to back translation, we reorder utterances by the SOTA reordering model SOW-REAP (Goyal and Durrett, 2020). To increase the coverage, we follow the same strategy as *Substitution (nf)* to generate an extended set of reordered utterances with multiple instances per input utterance. We use Sentence-Bert to encode sentences and select the top-k best ones according to their cosine similarity scores with the original input. The reordered sentences share the same vocabulary as the original one except for the order of words.

3.1.2 Filter Examples by Crowdsourcing

As perturbation operations related to function words rarely alter meanings, we apply three crowdsourcing operations to the utterances perturbed, including *Substitution (nf)*, *Reordering*, and *Back Translation*. For each perturbed utterance paired with the original utterance, three turkers at Amazon Mechanical Turk discern any semantic changes by choosing an option out of three — *the same*, *different*, or *not sure*. By default, any sentences uncomprehended by a human are regarded as *not sure*. Finally, we keep only the ones that have the same meaning agreed by at least two turkers. After crowdsourcing, we keep 83%, 82% and 61% of the generated utterances for *Substitution (nf)*, *Back Translation* and *Reordering*, respectively.

3.2 Evaluation Metrics

Our framework assesses the performance of a semantic parser w.r.t. standard accuracy and robustness metrics. Those robustness metrics indicate how well a semantic parser resists to meaning-preserving perturbations and adversarial attacks. As adversarial training is widely used for adversarial defense and mostly applicable to any neural semantic parsers, this framework supports comparing a wide range of adversarial training methods w.r.t. standard accuracy and robustness metrics.

A training set, a validation set, a standard test set, and a meaning-preserving test set are established in each domain. The first three sets are obtained from the original benchmark corpus, and the meaning-preserving test set is created using the methods described in Sec. 3.1. We will examine the meaning-preserving test set (denoted by $S_{\text{perturb}}(D)$) and its two subsets. We refer to the first subset as the robustness evaluation set (denoted by $R_{\text{eval}}(D)$), where the counterparts before perturbation are parsed correctly. We refer to the second subset as the black-box test set (denoted by $B_{\text{attack}}(D)$), where the loss of a parser to the examples is higher than their counterparts before perturbation. For each target parser, we consider four metrics, including *standard accuracy*, *perturbation accuracy*, *robust accuracy*, and *success rate of black-box attack*.

Standard accuracy The most widely used metric on semantic parsing (Dong and Lapata, 2018; Yin and Neubig, 2018) to measure the percentage of the predicted LFs that exactly match their gold LFs in a standard test set.

Perturbation accuracy Perturbation accuracy is formally defined as $n/|S_{\text{perturb}}(D)|$, where n denotes the number of correctly parsed examples to their gold LFs in a meaning-preserving test set $S_{\text{perturb}}(D)$.

Robust accuracy Robust accuracy is calculated as $n/|R_{\text{eval}}(D)|$, where n denotes the number of examples that are parsed correctly by a parser in a robustness evaluation set $R_{\text{eval}}(D)$. Compared to perturbation accuracy, robust accuracy measures the number of examples that a parser can parse correctly before perturbation but fails to get them right after perturbation.

Success rate of black-box attack A black-box attack example is regarded as the one that increases the loss of a model after perturbation (Zhang et al., 2020). Here, the success rate of black-box attack is calculated as $n/|B_{\text{attack}}(D)|$, where n denotes the number of examples that

are parsed *incorrectly* by a parser in the black-box test set $B_{\text{attack}}(D)$. White-box attacks require model specific implementation to generate adversarial examples, thus we leave the corresponding evaluation to the developers of semantic parsers.

The four metrics are computed to evaluate the efficacy of an adversarial training method. We inspect whether the metrics increase or decrease post-training and to what degree. An effective adversarial training method is expected to find a good trade-off between standard accuracy and robust accuracy.

Last but not least, all evaluation metrics are implemented with easy-to-use APIs in our toolkit. Our toolkit supports easy evaluation of a semantic parser and provides source code to facilitate integrating additional semantic parsing corpora.

4 Experiments

In this section, we present the first empirical study on robust semantic parsing.

4.1 Experimental Setup

Parsers We consider three SOTA neural semantic parsers — SEQ2SEQ with attention (Luong et al., 2015), COARSE2FINE (Dong and Lapata, 2018), and TRANX (Yin and Neubig, 2018). COARSE2FINE is the best performing semantic parser on the standard splits of GEOQUERY and ATIS. TRANX reports standard accuracy on par with COARSE2FINE and employs grammar rules to ensure validity of outputs.

Datasets Our evaluation corpus is constructed by extending three benchmark corpora — GEOQUERY (Zelle and Mooney, 1996), ATIS (Dahl et al., 1994), and JOBS. GEOQUERY contains 600 and 280 utterance-LF pairs to express the geography information in the training and test set, respectively. ATIS consists of 4434, 491, and 448 examples about flight booking in the training, validation, and test sets, respectively. And JOBS includes 500 and 140 pairs about job listing in the training and test set, respectively.

Adversarial Training Methods We apply three adversarial training methods to the parsers. We evaluate whether the three adversarial training methods could improve semantic parsers' robustness against the meaning-preserving examples generated by the word-level and sentence-level operations. The corresponding three adversarial training methods are as follows:

> **Fast Gradient Method (Miyato et al., 2016)** Fast Gradient Method (FGM) adds small perturbations to the word embeddings and train the semantic parsers with the perturbed embeddings. The perturbations are scaled gradients w.r.t. the input word embeddings.
>
> **Projected Gradient Descent (Madry et al., 2017)** Projected Gradient Descent (PGD) adds small perturbations to the word embeddings as well. Instead of calculating a single step of gradients, PGD accumulates the scaled gradients for multiple iterations to generate the perturbations.
>
> **Meaning-preserving Data Augmentation** Meaning-preserving Data Augmentation (MDA) augments the original training data with the meaning-preserving examples generated by the word-level and sentence-level operations. We randomly select 20% of the original instances for each dataset and generate their corresponding meaning-preserving instances. Since there are six different operations, each original training set is augmented with six different meaning-preserving sets independently.

Training Details For both supervised training and adversarial training, we set batch size to 20 for all parsers with 100 epochs. We follow the best settings of three parsers reported in (Dong and Lapata, 2018; Yin and Neubig, 2018) to set the remaining hyperparameters. The best performing implementation of SEQ2SEQ is included in Yin and Neubig (2018).

4.2 Results and Analysis

We discuss experimental results by addressing the following research questions:

RQ1: How do the SOTA parsers perform on meaning-preserving test sets? All three SOTA semantic parsers are trained on each training set before they are tested on the corresponding standard test sets and meaning-preserving test sets. Besides the overall results on whole test sets, Table 2 reports accuracy on each example subset perturbed by the respective perturbation operation. The results on those subsets are further compared

Generation Methods	Jobs			GeoQuery			Atis		
	Seq2Seq	Coarse2Fine	TranX	Seq2Seq	Coarse2Fine	TranX	Seq2Seq	Coarse2Fine	TranX
Insertion	87.09/84.98	84.40/77.77	81.89/82.97	85.59/80.50	87.57/85.21	85.96/81.73	67.36/68.33	84.16/82.73	77.45/75.63
Substitution (f)	86.58/78.89	84.22/71.59	82.23/86.10	85.27/78.38	86.97/82.30	84.23/79.04	70.93/59.19	86.06/68.22	76.70/67.69
Deletion	86.48/82.23	84.16/81.46	83.80/75.79	85.31/82.30	85.69/72.56	85.69/72.56	67.82/63.06	83.14/73.23	79.65/60.83
Substitution (nf)	80.34/78.63	79.05/76.92	82.90/83.33	85.71/64.95	87.37/70.93	86.37/66.11	75.28/56.36	85.57/68.94	78.55/60.57
Back Translation	87.66/44.15	84.41/46.10	87.01/58.44	85.92/48.74	86.93/56.28	85.42/41.13	75.91/56.38	86.55/63.30	81.23/58.36
Reordering	89.53/39.53	84.88/48.83	90.69/67.44	90.07/29.78	90.78/48.93	93.61/41.13	72.28/40.66	87.04/52.10	79.51/47.89
Overall(micro)	86.15/76.08	83.64/72.35	82.70/78.81	85.48/72.55	87.15/78.04	86.01/72.44	70.72/60.55	85.13/71.79	78.52/64.89
Overall(macro)	86.27/68.06	83.52/67.11	84.75/75.67	86.31/64.10	87.55/69.36	86.88/63.61	71.59/57.33	85.42/68.08	78.84/61.82

Table 2: Standard/Perturbation accuracy of SOTA parsers, trained on the original training sets.

Generation Methods	Jobs			GeoQuery			Atis		
	Seq2Seq	Coarse2Fine	TranX	Seq2Seq	Coarse2Fine	TranX	Seq2Seq	Coarse2Fine	TranX
Insertion	96.06	91.51	94.96	82.32	96.45	95.37	93.42	95.04	93.91
Substitution (f)	90.66	84.77	90.51	87.96	94.51	83.41	78.02	77.05	75.22
Deletion	94.64	96.33	97.18	88.03	93.67	92.04	88.04	86.06	8 7.03
Substitution (nf)	97.87	97.29	97.93	78.71	80.79	76.64	76.50	79.88	76.56
Back Translation	48.88	54.07	64.92	61.07	64.16	57.22	73.06	79.88	76.56
Reordering	42.85	57.53	73.06	38.21	53.90	44.96	53.75	58.82	57.63
Overall(micro)	87.79	86.03	92.21	83.08	89.06	85.05	81.29	82.27	83.06
Overall(macro)	78.50	80.08	86.43	72.31	80.58	74.94	77.71	78.12	76.82

Table 3: Robust accuracy for three SOTA parsers in each domain.

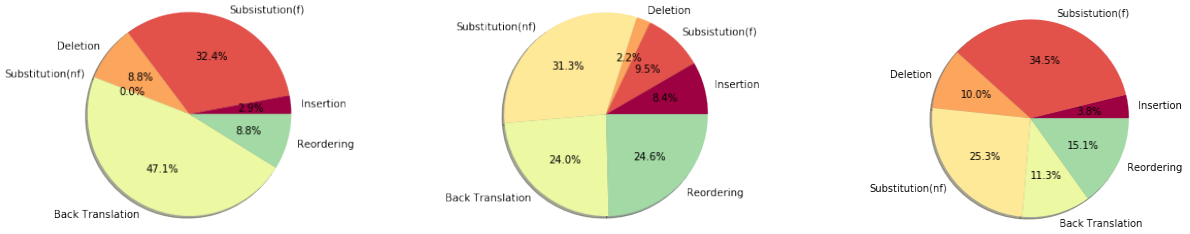
with the standard accuracy on the corresponding examples before perturbation.

As shown in Table 2, SOTA semantic parsers suffer from significant performance drop in almost all meaning-preserving test sets compared to the results on standard test sets. The performance ranking among the three SOTA semantic parsers varies across different datasets. Coarse2Fine achieves the best performance on GeoQuery and Atis, while Seq2Seq beats Coarse2Fine and TranX on Jobs. Although Coarse2Fine achieves better accuracy than TranX on the standard test set of Jobs, it falls short of TranX on the meaning-preserving test set of Jobs. A parser achieving higher standard accuracy does not necessarily obtain better perturbation accuracy against meaning-preserving perturbations than its competitors.

Our evaluation framework supports also in-depth analysis on the impact of different perturbation method on semantic parsers. All parsers are more vulnerable to sentence-level perturbations than word-level ones. Although reordering changes only word order in utterances, it leads to the lowest perturbation accuracy of all parsers on GeoQuery and Atis. Among word-level perturbation operations, substitution of non-function words is more challenging than deletion and insertion of function words on GeoQuery and Atis. On Jobs and Atis, even deletion or substitution of function words can impose significant challenges for the parsers. When we further investigate the deleted or replaced function words, such as in Table 1, semantic parsers are not expected to rely on such information. As all perburbations in our meaning-preserving test sets do not change meanings of original utterances, it imposes new research challenges on how to make semantic parsers resist meaning-preserving perturbations as well as how to avoid overfitting on semantically insignificant words.

RQ2: What kind of perturbed examples particularly degrade parser performance? Although perturbation accuracy allows comparing parsers on the same test sets, it includes the examples that a parser fails to parse correctly both before and after the meaning-preserving perturbation. Robust accuracy focuses on the examples a parser parses successfully before perturbation but fails after that. We investigate all parsers trained without adversarial training in terms of this measure. As shown in Table 3, TranX is superior to the other two parsers on Jobs, and Coarse2Fine is the clear winner on GeoQuery. This ranking of parsers is consistent with perturbation accuracy in the two domains. However, the differences among three parsers on Atis are marginal, while Coarse2Fine achieves significantly superior perturbation accuracy in the same domain. Atis is the domain with the most diverse paraphrases in natural language, Coarse2Fine cannot significantly outperform the other two parsers against meaning-preserving perturbations.

We further investigate adversary examples, which are defined in Eq. 2, for each parser in the meaning-preserving test sets. The shared adversarial examples among the parsers vary significantly across domains. More than 50% of the ad-



(a) JOBS (b) GEOQUERY (c) ATIS

Figure 1: Contributions of perturbation operations on the shared adversaries.

versarial examples are shared among the parsers on ATIS, but only less than 25% of adversaries are shared on JOBS. Fig. 1 illustrates which perturbation operation contributes to the intersection of the adversary set from different parsers. Back translation contributes almost half of the shared perturbed examples on JOBS, while the proportion of word-level perturbation operations reaches nearly 70% on ATIS. Although reordering and back translation impose the most difficult challenges for parsers, they cannot generate a significant number of valid adversarial examples. In contrast, it is relatively easy to resist a word substitution attack, but this operation can be easily applied to generate a large number of adversaries.

Insertion	
Utterance	show me the nonstop flight and fare from toronto to st. petersburg
Adv. Utterance	show me the nonstop flight and fare from from toronto to st. petersburg
Ground Truth	(lambda $0 e (and (flight $0) (nonstop $0) (from $0 toronto:ci) (to $0 st.petersburg:ci)))
SEQ2SEQ	(lambda $0 e (exists $1 (and (flight $0) (nonstop $0) (from $0 toronto:ci) (to $0 st.petersburg:ci) (= (fare $0) $1))))
COARSE2FINE	(lambda $0 (exists $1 (and (flight $0) (nonstop $0) (from $0 toronto:ci) (to $0 st.petersburg:ci) (= (fare $0) $1))))
TRANX	(lambda $0 e (exists $1 (and (flight $0) (nonstop $0) (from $0 toronto:ci) (to $0 st.petersburg:ci) (= (fare $0) $1))))
Substitution(f)	
Utterance	what is the capital of the state with the largest population density
Adv. Utterance	what are the capital of the state with the largest population density
Ground Truth	(capital:c (argmax $1 (state:t $1) (density:i $1)))
SEQ2SEQ	(lambda $0 e (and (capital:t $0) (loc:t $0 (argmax $1 (state:t $1) (density:i $1)))) (density:i $0))
COARSE2FINE	(argmax $0 (state:t $0) (density:i $0))
TRANX	(lambda $0 e (and (capital:t $0) (loc:t $0 (argmax $1 (state:t $1) (density:i $1)))))
Deletion(f)	
Utterance	what is the capital of the state with the largest population density
Adv. Utterance	what is the capital ~~of~~ the state with the largest population density
Ground Truth	(capital:c (argmax $1 (state:t $1) (density:i $1)))
SEQ2SEQ	(argmin $0 (capital:t $0) (density:i $0))
COARSE2FINE	(argmax $0 (capital:t $0) (population:i $0))
TRANX	(argmax $0 (capital:t $0) (density:i $0))
Substitution(nf)	
Utterance	what are the major river in ohio
Adv. Utterance	what are the main river in ohio
Ground Truth	(lambda $0 e (and (major:t $0) (river:t $0) (loc:t $0 ohio)))
SEQ2SEQ	(lambda $0 e (and (river:t $0) (loc:t $0 ohio)))
COARSE2FINE	(lambda $0 (and (river:t $0) (loc:t $0 ohio)))
TRANX	(lambda $0 e (and (river:t $0) (loc:t $0 ohio)))
Back Translation	
Utterance	list job using sql
Adv. Utterance	what kind of work do you have with sql
Ground Truth	job (ANS) , language (ANS , 'sql')
SEQ2SEQ	job (ANS) , application (ANS , 'sql')
COARSE2FINE	job (ANS) , language (ANS 'sql') , language (ANS 'sql')
TRANX	job (ANS) , language (ANS 'sql') , language (ANS 'sql')
Reordering	
Utterance	what is the lowest point in mississippi
Adv. Utterance	in s0 , the lowest point in mississippi
Ground Truth	(lambda $0 e (loc:t (argmin $1 (and (place:t $1) (loc:t $1 mississippi:s)) (elevation:i $1)) $0))
SEQ2SEQ	(argmin $0 (and (place:t $0) (loc:t $0 mississippi:s)) (elevation:i $0))
COARSE2FINE	(lambda $0 (loc:t mississippi:s $0) (loc:t (argmin $1 (and (place:t $1) (loc:t $1 mississippi:s)) (elevation:i $1)) $0))
TRANX	(argmin $0 (and (place:t $0) (loc:t $0 mississippi:s)) (elevation:i $0))

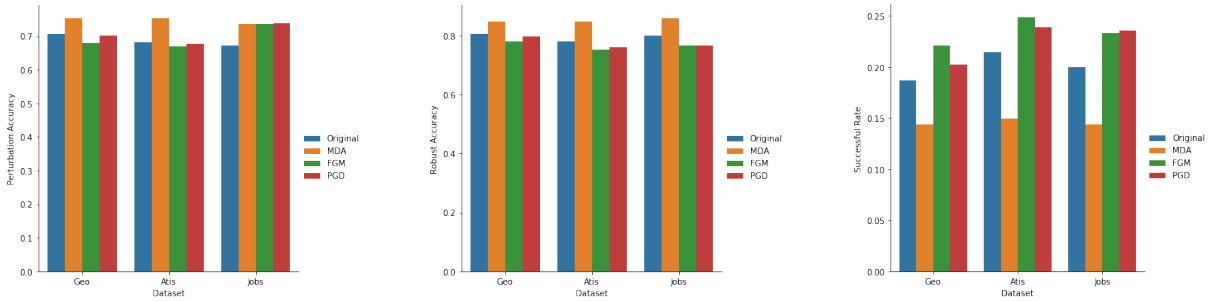
Table 4: Adversarial examples shared by all parsers.

We hand picked representative adversarial examples shared by all parsers and list them in Table 4. The errors made by substitution (f) and insertion often show a sign of overfitting due to the fact that the parsers learn dependencies between non-essential features and predicates. Substitution (nf) sometimes causes predicates to be missing after replacing a word with its synonym. The errors caused by deletion, back translation, and reordering are more diverse, including adding wrong predicates or missing predicates.

RQ3: How effective are those widely used adversarial training methods for semantic parsers? We applied FGM, PGD, and MDA to train the three parsers on the respective training sets. In the case of MDA, we consider all automatic perturbation operations to generate training examples, and each operation adds the same amount of training examples to the original train sets. Fig. 2 illustrates the results of COARSE2FINE w.r.t. different evaluation metrics. MDA always improves the three evaluation metrics, despite that the generated training examples are not checked by humans, resulting in a significant number of meaning-changing examples. FGM and PGD hardly work except on JOBS in terms of perturbation accuracy. Since both methods are white-box methods, the results indicate the gradient-based adversaries cannot effectively capture the meaning-preserving perturbations.

The biggest improvement made by MDA is the success rate of a black-box attack. The black-box attack examples are selected from the meaning-preserving examples that increase a parser's loss after perturbation. When we inspect the examples in the meaning-preserving test sets generated by different perturbation operation, the biggest improvement is from the ones perturbed by sentence-level operations and word substitution. All three parsers benefit from MDA significantly. The more vulnerable to black-box attack a parser is, the more improvement it can achieve after applying MDA.

RQ4: How does each adversarial training method influence standard accuracy? Tsipras et al. (2018) point out that there is a trade-off be-



(a) Perturbation accuracy. (b) Robust accuracy. (c) Success rate of black-box attack.

Figure 2: Robustness metrics of COARSE2FINE before and after adversarial training.

Generation Methods	JOBS			GEOQUERY			ATIS		
	SEQ2SEQ	COARSE2FINE	TRANX	SEQ2SEQ	COARSE2FINE	TRANX	SEQ2SEQ	COARSE2FINE	TRANX
No adv. train	87.14	83.57	87.14	83.92	86.78	86.07	73.88	86.16	79.91
FGM	87.31	86.12	91.34	83.89	86.35	84.01	81.00	79.81	81.72
PGD	82.84	86.18	92.93	84.56	86.35	84.01	79.96	79.94	81.52
MDA(all)	87.74	86.16	88.57	82.89	86.73	87.11	74.75	80.18	80.60
MDA(Insertion)	89.14	86.16	88.85	81.78	87.14	87.64	67.16	85.27	78.75
MDA(Substitution (f))	87.71	85.25	88.85	83.21	86.79	87.35	76.51	78.08	85.67
MDA(Deletion)	87.28	90.14	85.33	83.21	86.79	87.35	76.87	85.62	78.52
MDA(Substitution (nf))	87	80.68	90.14	83.85	86.79	86.85	76.11	62.58	85.58
MDA(Back Translation)	88.42	87.53	88.57	82.14	86.71	86.64	76.20	84.73	77.36
MDA(Reordering)	86.91	87.21	89.71	83.21	86.21	86.85	75.66	84.82	77.72

Table 5: Standard accuracy of adversarial training, which is trained with adversarial training methods and evaluated on the standard test sets. MDA(all) stands for MDA with all six perturbation operations.

tween standard accuracy and robust accuracy in most occasions. This finding is attributed to the presence of *non-robust* features, which are highly predictive but incomprehensible for humans (Ilyas et al., 2019). To verify the theory, after applying each adversarial training method to each parser, we compare the standard accuracy before and after adversarial training.

As shown in Table 5, none of the three adversarial training methods consistently improve standard accuracy across different domains and parsers. TRANX does not have a significant performance drop regardless which adversarial training method is applied. It may be due to the fact that TRANX uses grammar to filter out invalid outputs.

MDA cannot consistently improve parsers but also does not hurt parsers' performance in terms of standard accuracy. We conducted t-tests on the standard test sets to assess if MDA with different perturbation operations significantly improves accuracy. The results are negative so that the training examples generated by MDA at least do not hurt parsers' performance while increasing their robustness.

RQ5: How does our meaning-preserving data augmentation method compare with the data augmentation method proposed by (Jia and Liang, 2016)? (Jia and Liang, 2016) is one of the SOTA data augmentation methods for semantic parsing. In their work, they show that the augmented examples improve accuracy of predicting denotations. Although there are three methods proposed in (Jia and Liang, 2016), only the method of *concatenating* two random examples as a new example can be applied in our case. We evaluated this augmentation method with all three parsers. No significant improvement of standard accuracy and robust accuracy is found in all three domains. We conjecture that this is due to the fact that in (Jia and Liang, 2016) they report only improvement of accuracy of denotation matching, not the matching of LFs. In contrast, MDA can effectively reduce the harm of meaning-preserving perturbations.

4.3 Conclusion

We conduct the empirical study on robustness of neural semantic parsers. In order to evaluate robustness accuracy, we define first what are adversarial examples for semantic parsing, followed by constructing test sets to measure robustness using a scalable method. The outcome of this work is supposed to facilitate semantic parsing research by providing a benchmark for evaluating both standard accuracy and robustness metrics.

References

Anish Athalye and Nicholas Carlini. 2018. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Matthias Blohm, Glorianna Jagfeld, Ekta Sood, Xiang Yu, and Ngoc Thang Vu. 2018. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. *CoRR*, abs/1808.08744.

Alvin Chan, Lei Ma, Felix Juefei-Xu, Xiaofei Xie, Yang Liu, and Yew-Soon Ong. 2018. Metamorphic relation based adversarial attacks on differentiable neural computer. *CoRR*, abs/1809.02444.

Bo Chen, Le Sun, and Xianpei Han. 2018. Sequence-to-action: End-to-end semantic graph generation for semantic parsing. *arXiv preprint arXiv:1809.00773*.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *AAAI*, pages 3601–3608.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, page 43–48, USA. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. *CoRR*, abs/1806.09030.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for NLP. *CoRR*, abs/1712.06751.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. *arXiv preprint arXiv:2005.02013*.

Chuan Guo, Jacob R Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q Weinberger. 2019a. Simple black-box adversarial attacks. *arXiv preprint arXiv:1905.07121*.

Daya Guo, Yibo Sun, Duyu Tang, Nan Duan, Jian Yin, Hong Chi, James Cao, Peng Chen, and Ming Zhou. 2018. Question generation from sql queries improves neural semantic parsing. *arXiv preprint arXiv:1808.06304*.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019b. Towards complex text-to-sql in cross-domain database with intermediate representation. *CoRR*, abs/1905.08205.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. *arXiv preprint arXiv:1803.02400*.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *CoRR*, abs/1812.00978.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1033–1043, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. *arXiv preprint arXiv:1904.05780*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

A. Madry, Aleksandar Makelov, L. Schmidt, D. Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Paul Michel, Xian Li, Graham Neubig, and Juan Miguel Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. *arXiv preprint arXiv:1903.06620*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. *CoRR*, abs/1604.08275.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2018. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

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.

Wenqi Wang, Benxiao Tang, Run Wang, Lina Wang, and Aoshuang Ye. 2019. A survey on adversarial attacks and defenses in text. *CoRR*, abs/1902.07285.

Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. 2020. Elephant in the room: An evaluation framework for assessing adversarial examples in nlp. *arXiv preprint arXiv:2001.07820*.

Pengcheng Yin and Graham Neubig. 2018. Tranx: A transition-based neural abstract syntax parser for semantic parsing and code generation.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transaction on Intelligent System and Technology*, 11(3).