Don't paraphrase, detect! Rapid and Effective Data Collection for Semantic Parsing

Jonathan Herzig[1] **Jonathan Berant**[1,2]

[1]School of Computer Science, Tel-Aviv University
[2]Allen Institute for Artificial Intelligence
{jonathan.herzig,joberant}@cs.tau.ac.il

Abstract

A major hurdle on the road to conversational interfaces is the difficulty in collecting data that maps language utterances to logical forms. One prominent approach for data collection has been to automatically generate pseudo-language paired with logical forms, and paraphrase the pseudo-language to natural language through crowdsourcing (Wang et al., 2015). However, this data collection procedure often leads to low performance on real data, due to a mismatch between the true distribution of examples and the distribution induced by the data collection procedure. In this paper, we thoroughly analyze two sources of mismatch in this process: the mismatch in *logical form distribution* and the mismatch in *language distribution* between the true and induced distributions. We quantify the effects of these mismatches, and propose a new data collection approach that mitigates them. Assuming access to unlabeled utterances from the true distribution, we combine crowdsourcing with a paraphrase model to detect correct logical forms for the unlabeled utterances. On two datasets, our method leads to 70.6 accuracy on average on the *true distribution*, compared to 51.3 in paraphrasing-based data collection.

1 Introduction

Conversing with a virtual assistant in natural language is one of the most exciting current applications of semantic parsing, the task of mapping natural language utterances to executable logical forms (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Liang et al., 2011). Semantic parsing models rely on supervised training data that pairs natural language utterances with logical forms. Alas, such data does not occur naturally, especially in virtual assistants that are meant to support thousands of different applications and use-cases. Thus, efficient data collection is perhaps the most pressing problem for scalable conversational interfaces.

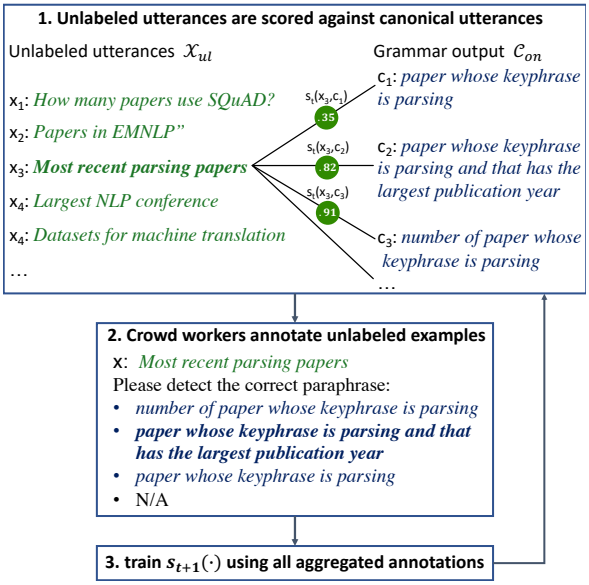

Figure 1: An overview of GRANNO, a method for annotating unlabeled utterances with their logical forms.

In recent years, many attempts aim to reduce the burden of data collection for semantic parsing, including training from denotations (Kwiatkowski et al., 2013; Artzi and Zettlemoyer, 2013), semi-supervised learning (Kociský et al., 2016; Yin et al., 2018), human in the loop (Iyer et al., 2017; Lawrence and Riezler, 2018), and training on examples from other domains (Herzig and Berant, 2017, 2018; Su and Yan, 2017). One prominent approach for data collection was introduced by Wang et al. (2015), termed OVERNIGHT. In this method, one automatically generates logical forms for an application from a grammar, paired with pseudo-language utterances. These pseudo-language utterances are then shown to crowd workers, who are able to understand them and paraphrase them into natural language, resulting

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3810–3820, Hong Kong, China, November 3–7, 2019.

in a supervised dataset.

The OVERNIGHT procedure has been adopted and extended both within semantic parsing (Locascio et al., 2016; Ravichander et al., 2017; Zhong et al., 2017; Cheng et al., 2018), in dialogue (Shah et al., 2018; Damonte et al., 2019) and in visual reasoning (Johnson et al., 2017; Hudson and Manning, 2019).

While the OVERNIGHT approach is appealing since it generates training data from scratch, it suffers from a major drawback – training a parser on data generated from OVERNIGHT and testing on utterances collected from a target distribution results in a significant drop in performance (Wang et al., 2015; Ravichander et al., 2017).

In this paper, we thoroughly analyze the sources of mismatch between a target distribution and the distribution induced by OVERNIGHT, and propose a new method for overcoming this mismatch. We decouple the mismatch into two terms: the *logical form mismatch*, i.e., the difference between the target distribution of logical forms and the distribution obtained when generating from a grammar, and the *language mismatch*, i.e., the difference between the natural language obtained when paraphrasing a pseudo-language utterance and the language obtained by real users of an application.

We find that the most severe problem arising from the *logical form mismatch* is insufficient coverage of logical form templates that occur in the true distribution, when generating from a grammar. We also isolate the *language mismatch* effect by paraphrasing logical forms sampled from the true logical form distribution, and find that the language mismatch alone results in a decrease of 9 accuracy points on average on two datasets.

To overcome these mismatches, we propose an alternative method to OVERNIGHT, that utilizes unlabeled utterances. Our method, named GRANNO (grammar-driven annotation), allows crowd workers to iteratively annotate unlabeled utterances by *detecting* their correct grammar generated paraphrase. Figure 1 illustrates a single iteration of this approach. An unlabeled utterance is matched using a paraphrase model against candidate pseudo-language utterances generated by the grammar (step 1). The unlabeled utterance and its top candidate paraphrases are presented to a crowd worker that detects the correct paraphrase (step 2). The paraphrase model is re-trained given all annotated utterances thus far (step 3), and is used to score the remaining unlabeled utterances.

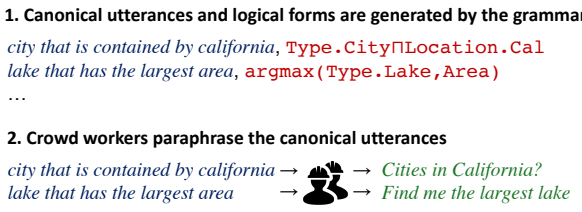

Figure 2: OVERNIGHT: Canonical utterances are generated by a grammar and paraphrased by crowd workers.

On two semantic parsing datasets, we show our procedure leads to annotation of 89% of the original training data. The accuracy of the resulting parser is 70.6 on average, well beyond the accuracy obtained through the original OVERNIGHT procedure at 51.3. This substantially closes the gap to a fully-supervised semantic parser, which obtains 84.9 accuracy. All our code and collected data is available at https://github.com/jonathanherzig/semantic-parsing-annotation.

2 The OVERNIGHT Framework

We now describe the OVERNIGHT framework for data collection, which we investigate and improve upon in this work. The starting point is a user who needs a semantic parser for some domain, but has *no data*. OVERNIGHT describes a procedure for generating training data from scratch, which comprises two steps (see Figure 2). First, a synchronous grammar is used to generate logical forms paired with *canonical utterances*, which are pseudo-language utterances that are understandable to people, but do not sound natural. Second, crowd workers paraphrase these canonical utterances into natural language utterances. This results in a training set of utterances paired with logical forms that is used to train the semantic parser. We now briefly elaborate on these two steps.

Grammar The grammar in OVERNIGHT generates logical forms paired with canonical utterances, which are understandable to crowd workers (e.g., *"number of state that borders california"*). The grammar has two parts: the *domain-*

general part contains domain-independent rules that cover logical operators (e.g., comparatives, superlatives, negation etc.). In addition, a *seed lexicon* specifies a canonical phrase (*"publication year"*) for each knowledge-base (KB) constant (`PublicationYear`), in each particular domain.

While there are many possible ways to sample data from the grammar, in OVERNIGHT logical forms and canonical utterances are exhaustively generated up to a certain maximal depth, hereafter denoted D. This follows the assumption that the semantic parser, trained from this data, should generalize to logical forms that correspond to deeper trees (we re-examine this assumption in Section 3). In addition, a typing system is used during generation, thus, semantically vacuous logical forms are not generated (e.g., `PublicationYear.Parsing`), which substantially reduces the number of generated examples.

Crowdsourcing After the above generation procedure terminates, crowd workers paraphrase each canonical utterance into a natural language utterance (e.g., the canonical utterance *"paper that has the largest publication year"* can be paraphrased to *"what is the most recent paper?"*). Finally, the framework yields a training dataset $\mathcal{D}_{\text{on}} = \{(x_i, z_i)\}_{i=1}^{N_{\text{on}}}$ consisting of pairs of natural language utterances and logical forms, which can be used to train a semantic parser.

3 Mismatch Analysis

In supervised semantic parsing, we assume access to training data of the form $\mathcal{D}_{\text{nat}} = \{(x_i, z_i)\}_{i=1}^{N_{\text{nat}}}$ sampled from the true target distribution $p_{\text{nat}}(x, z)$. Conversely, in the OVERNIGHT framework, we train from $\mathcal{D}_{\text{on}}$, which is sampled from a different distribution $p_{\text{on}}(x, z)$. Training on data sampled from $p_{\text{on}}(x, z)$ and testing on data sampled from $p_{\text{nat}}(x, z)$ leads to a substantial decrease in performance (compared to training on data sampled from $p_{\text{nat}}(x, z)$). In this section, we will analyze and quantify the causes for this degradation in performance.

By writing $p_{\text{nat}}(x, z) = p_{\text{nat}}(z)p_{\text{nat}}(x \mid z)$ and $p_{\text{on}}(x, z) = p_{\text{on}}(z)p_{\text{on}}(x \mid z)$ (mirroring the data generation procedure of OVERNIGHT), we can decouple the distribution mismatch into two terms: The first is the *logical form mismatch*, i.e., the difference between the natural distribution of logical forms $p_{\text{nat}}(z)$ and the distribution $p_{\text{on}}(z)$ of logical forms induced by OVERNIGHT. The second is the *language mismatch*, i.e., the difference between the conditional distribution of natural language $p_{\text{nat}}(x \mid z)$ and the conditional distribution $p_{\text{on}}(x \mid z)$ of natural language when performing crowdsourcing with the OVERNIGHT procedure. We will now investigate these two types of mismatch and their interaction with neural semantic parsers in the context of two popular semantic parsing datasets, GEOQUERY (Zelle and Mooney, 1996) and SCHOLAR (Iyer et al., 2017), which focus on the domains of geography and publications, respectively. For this analysis, we replicated the OVERNIGHT procedure and generated a dataset $\mathcal{D}_{\text{on}}$ for these two domains (details in Section 5).

3.1 Logical Form Mismatch

The first source of mismatch is that the natural distribution of logical forms $p_{\text{nat}}(z)$ can be quite different from the distribution $p_{\text{on}}(z)$ induced by the ad-hoc procedure for generating logical forms in OVERNIGHT.

Logical Operators Frequency A simple way to quantify the logical form mismatch, is to look whether the proportion of different logical operators, such as superlatives (e.g., `argmax`) and comparatives (e.g., `>`), substantially differs between $\mathcal{D}_{\text{on}}$ and $\mathcal{D}_{\text{nat}}$. Table 1 examines this, where hereafter we use the original training and development sets of GEOQUERY and SCHOLAR as $\mathcal{D}_{\text{nat}}$. The table shows that the frequency of logical operators is different between the two training sets $\mathcal{D}_{\text{nat}}$ and $\mathcal{D}_{\text{on}}$. For example, in $\mathcal{D}_{\text{nat}}$ of GEOQUERY, only 1% of the examples involve negation, compared to 55% of the examples in $\mathcal{D}_{\text{on}}$. Such differences may present generalization difficulties for a model trained from $\mathcal{D}_{\text{on}}$, as the real distribution we care about is $\mathcal{D}_{\text{nat}}$.

Coverage Wang et al. (2015) proposed to exhaustively generate all logical forms corresponding to a derivation tree of maximal depth D, assuming that the model will generalize to logical forms that require deeper trees. However, recent work from Finegan-Dollak et al. (2018) showed this might not be the case. In their work, Finegan-Dollak et al. (2018) consider logical form *templates*, i.e., logical forms where the KB constants are abstracted to their semantic type, and split the data such that templates in the test set are disjoint from templates in the training set. They show that neural semantic parsers struggle to generalize in this setup. Thus, *covering* the logical form

operator	GEOQUERY		SCHOLAR		example
	$\mathcal{D}_{\text{nat}}$	$\mathcal{D}_{\text{on}}$	$\mathcal{D}_{\text{nat}}$	$\mathcal{D}_{\text{on}}$	
argmax	.27	.14	.18	.08	*"the longest river"*
argmin	.10	.14	.02	.08	*"earliest paper by..."*
larger	.01	.06	.02	.10	*"high point higher than that of..."*
smaller	.00	.06	.02	.10	*"paper with less than 10 citations"*
conj_1	.81	.20	.50	.22	*"parsing papers"*
conj_2	.19	.80	.42	.78	*"2019 parsing papers"*
conj_3	.00	.00	.06	.00	*"2019 EMNLP parsing papers"*
conj_4	.00	.00	.01	.00	*"2019 EMNLP parsing papers by..."*
negation	.01	.55	.01	.57	*"state with no rivers"*
aggregation	.01	.04	.01	.03	*"total population of all states"*
count	.10	.02	.16	.02	*"number of papers published by..."*

Table 1: Logical operators frequency within the total number of examples. conj_n refers to a logical from that contains a conjunction of n clauses.

Dataset	D	$\lvert\mathcal{D}_{\text{on}}\rvert$	$\mathcal{D}_{\text{nat}}$ coverage
SCHOLAR	7	655,516	95.16
	6	91,739	90.85
	5	2,283	63.73
	4	91	17.44
GEOQUERY	7	2,086,830	94.13
	6	249,617	89.43
	5	4,480	76.52
	4	166	51.68

Table 2: Logical form coverage of $\mathcal{D}_{\text{nat}}$ by $\mathcal{D}_{\text{on}}$, after converting logical forms to their templates, for different values of D.

templates that appear in $\mathcal{D}_{\text{nat}}$ is important in the OVERNIGHT procedure.

Table 2 shows the number of examples generated by OVERNIGHT for different values of D, along with the proportion of examples in $\mathcal{D}_{\text{nat}}$ whose logical form template is covered by $\mathcal{D}_{\text{on}}$. Because OVERNIGHT requires paraphrasing each generated example, a reasonable value for D is 5. We observe that in this case only 63.73% of the examples in SCHOLAR are covered, as well as 76.52% of the examples in GEOQUERY.

To verify whether coverage is indeed important in our setup, we performed the following experiment. We trained a semantic parser (detailed in Section 5) on $\mathcal{D}_{\text{on}}$ for GEOQUERY and SCHOLAR (with D$=5$), and evaluated its performance on the training set of $\mathcal{D}_{\text{nat}}$ (which serves as a development set in this context). Then, we calculated the accuracy of the model with respect to examples for which their template appears in $\mathcal{D}_{\text{on}}$ (acc_{cov}), and for those their template does not appear (acc_{disj}). Table 3 shows that acc_{disj} is substantially lower

Dataset	acc_{cov}	acc_{disj}
GEOQUERY	78.73	10.71
SCHOLAR	65.58	5.24

Table 3: Denotation accuracy for the set of examples in $\mathcal{D}_{\text{nat}}$ covered by $\mathcal{D}_{\text{on}}$ (acc_{cov}), and for the set of uncovered examples (acc_{disj}).

than acc_{cov} for both datasets. This shows the importance of generating logical form templates that are likely to appear in the target distribution. Thus, our finding reinforces the conclusions in (Finegan-Dollak et al., 2018), and shows that while neural semantic parsers obtain good performance on existing benchmarks, their generalization to new compositional structures is limited.

Unlikely Logical Forms In OVERNIGHT, logical forms that are unlikely to appear in a natural interaction could be generated. For instance, the canonical utterance *"total publication year of paper whose keyphrase is semantic parsing"*, which refers to the sum of publication years of all semantic parsing papers. Although such logical forms are valid with respect to type matching, users are unlikely to produce them, and they are hard to prune automatically. To estimate the logical form mismatch caused by unlikely logical forms, we manually inspected 100 random examples from $\mathcal{D}_{\text{on}}$ with D$=5$ for GEOQUERY, and found that 31% of the examples are unlikely. While these examples are not necessarily harmful for a model, they are difficult to paraphrase, and may introduce noisy paraphrases that hurt performance.

Example 1
c: *"river that traverses california and that has the largest length"*
x_{on}: *"What river that traverses California has the largest length?"*
x_{nat}: *"What is the longest river in California?"*

Example 2
c: *"paper whose keyphrase is deep learning and that has the largest publication year"*
x_{on}: *"Name the paper with a deep learning keyphrase that has the most recent publication year."*
x_{nat}: *"Most recent deep learning papers."*

Figure 3: Examples for canonical utterances (c) generated by the grammar, their paraphrase by crowd workers (x_{on}), and their natural utterance in $\mathcal{D}_{\text{nat}}$ (x_{nat}). While the paraphrases are correct, they are biased towards the language style in c.

3.2 Language mismatch

The second mismatch in OVERNIGHT, demonstrated in Figure 3, is between the language used when naturally interacting with a conversational interface, and the language crowd workers use when elicited to generate paraphrases conditioned on a canonical utterance.

To directly measure the language mismatch, we performed the following experiment. We generated a dataset $\mathcal{D}_{\text{lang}}$ by taking the examples from $\mathcal{D}_{\text{nat}}$ and paraphrasing their logical forms using the OVERNIGHT procedure. This ensures that $p_{\text{lang}}(z) = p_{\text{nat}}(z)$, and thus the only difference between $\mathcal{D}_{\text{nat}}$ and $\mathcal{D}_{\text{lang}}$ is due to *language mismatch*. Then we measured the difference in performance when training on these two datasets.

In more detail, for every example in $\mathcal{D}_{\text{nat}}$, we extracted the corresponding canonical utterance template (since the examples are covered by the OVERNIGHT grammar), denoted by $\mathcal{C}_{\text{nat}}$. This list of canonical utterances can be viewed as an oracle output of the OVERNIGHT grammar generation procedure, since they do not exhibit any logical form mismatch. Next, 12 NLP graduate students paraphrased the examples in $\mathcal{C}_{\text{nat}}$. The students were presented with guidelines and examples for creating paraphrases, similar to the original guidelines in Wang et al. (2015). Moreover, we explicitly asked to paraphrase the canonical utterances such that the output is significantly different from the input (while preserving the meaning). After the paraphrase process was completed, we manually fixed typos in all generated paraphrases. Thus, our paraphrasing procedure yields high quality paraphrases and is an upper bound to what can be achieved by crowd workers.

Model	GEOQUERY	SCHOLAR
SUPERVISED+ELMO	86.3	83.4
OVERNIGHT-ORACLE-LF+ELMO	77.7	73.5
SUPERVISED+GLOVE	84.9	78.2
OVERNIGHT-ORACLE-LF+GLOVE	71.6	67.2

Table 4: Denotation accuracy on the test set, comparing a semantic parser trained on $\mathcal{D}_{\text{nat}}$ (SUPERVISED) and parser trained on $\mathcal{D}_{\text{lang}}$ (OVERNIGHT-ORACLE-LF).

We trained a neural semantic parser on both $\mathcal{D}_{\text{nat}}$ (named SUPERVISED) and $\mathcal{D}_{\text{lang}}$ (named OVERNIGHT-ORACLE-LF) using the original train/development split. Results in Table 4 show that for each domain, a decrease of approximately 9 points in accuracy occurs only due to the language mismatch, even with high-quality workers. This gap is likely to grow when workers are not experienced in paraphrasing, or are unfamiliar with the domain they generate paraphrases for.

We additionally observe that when we use GloVe embeddings (Pennington et al., 2014) instead of contextual ELMo embeddings (Peters et al., 2018), the gap is even higher. This shows that better representations reduce the language gap, though it is still substantial.

4 Grammar-driven Annotation

To overcome the logical form and language mismatches discussed, we propose in this section a new data generation procedure that does not suffer from these shortcomings. Our procedure, named GRANNO, relies on two assumptions. First, unlike OVERNIGHT, we assume access to unlabeled utterances $\mathcal{X}_{\text{ul}} = \{x_i\}_{i=1}^{N_{\text{ul}}}$. These can typically be found in query logs, or generated by users experimenting with a prototype. Second, we assume a scoring function $s_0(x, c)$, which provides a reasonable initial similarity score between a natural language utterance x and a canonical utterance c.

The goal of our procedure is to iteratively label $\mathcal{X}_{\text{ul}}$ with crowd workers, aided by the OVERNIGHT grammar. If we manage to label most of the dataset $\mathcal{X}_{\text{ul}}$, which is sampled from the true target distribution, we will end up with a labeled dataset $\mathcal{D}_{\text{ga}}$ that has very little distribution mismatch.

Figure 1 illustrates the procedure. First, we generate all canonical utterances $\mathcal{C}_{\text{on}}$ up to a depth D from the OVERNIGHT grammar. Because we

Algorithm 1 GRANNO

Input: unlabeled utterances $\mathcal{X}_{\text{ul}}$, grammar G
Output: $\mathcal{D}_{\text{ga}}$ - training data for a semantic parser

generate $\mathcal{C}_{\text{on}}$ from G
$\mathcal{D}_{\text{ga}} \leftarrow \emptyset$, $s_0(\cdot) \leftarrow -WMD(\cdot)$, $t \leftarrow 0$
converge $\leftarrow False$, $\mathcal{S}_{\text{pos}} \leftarrow \emptyset$, $\mathcal{S}_{\text{neg}} \leftarrow \emptyset$
while not *converge* **do**
  for x **in** $\mathcal{X}_{\text{ul}}$ **do**
    calculate $\mathcal{C}_x^K$ using $s_t(\cdot)$ ▷ top K candidates
    crowd workers annotate c_x given x and $\mathcal{C}_x^K$
    if $c_x \neq N/A$ **then**
      $\mathcal{D}_{\text{ga}} \leftarrow \mathcal{D}_{\text{ga}} \cup (x, c_x)$
      $\mathcal{X}_{\text{ul}} \leftarrow \mathcal{X}_{\text{ul}} \setminus x$
      $\mathcal{S}_{\text{pos}} \leftarrow \mathcal{S}_{\text{pos}} \cup (x, c_x)$ ▷ positive examples
      for c **in** $\mathcal{C}_x^M \setminus \{c_x\}$ **do**
        $\mathcal{S}_{\text{neg}} \leftarrow \mathcal{S}_{\text{neg}} \cup (x, c)$ ▷ negative examples
  converge $\leftarrow$ check for convergence, $t \leftarrow t + 1$
  train $s_t(\cdot)$ over $\mathcal{S}_{\text{pos}}$ and $\mathcal{S}_{\text{neg}}$
return $\mathcal{D}_{\text{ga}}$

do not paraphrase all the canonical utterances, we can generate to a higher depth D compared to OVERNIGHT and cover more of the examples in $\mathcal{X}_{\text{ul}}$ (see Table 2). Then, we iteratively label the utterances in $\mathcal{X}_{\text{ul}}$. At each iteration t, we use a paraphrase detection model $s_t(x, c)$ to present promising canonical utterances to crowd workers for each unlabeled utterance x, who label the dataset. Importantly, crowd workers in our setup do not *generate paraphrases*, they only *detect* them. We now describe GRANNO in more detail.

Iterative Annotation At each iteration t, we rely on a trained scoring function $s_t(x, c)$ that estimates the similarity between an unlabeled utterance $x \in \mathcal{X}_{\text{ul}}$ and a generated canonical utterance $c \in \mathcal{C}_{\text{on}}$. We follow the procedure described in Algorithm 1. For an utterance $x \in \mathcal{X}_{\text{ul}}$, we calculate the top-$K$ ($= 5$) most similar canonical utterances in $\mathcal{C}_{\text{on}}$, denoted by $\mathcal{C}_x^K$. We then present x along with its candidate paraphrases $\mathcal{C}_x^K$ to a worker, and ask her to choose the correct candidate paraphrase. If a paraphrase does not appear in the top-K candidates, the worker selects no candidates.

These annotations are then used to train $s_{t+1}(x, c)$, which will be used in the next iteration. For each $x \in \mathcal{X}_{\text{ul}}$ for which a worker detected a paraphrase c_x, we label (x, c_x) as a positive example. We use the top-M ($= 100$) other most similar canonical utterances $\mathcal{C}_x^M \setminus \{c_x\}$ (according to $s_t(\cdot)$) as negative examples. We train s_{t+1} from all the examples generated in iterations $0 \cdots t$. Thus, in every iteration more and more examples are labeled, and a better scoring function is trained. We stop when we meet convergence, i.e., when no new unlabeled utterances are labeled. We then output the dataset $\mathcal{D}_{\text{ga}}$ that contains every utterance x from $\mathcal{X}_{\text{ul}}$ that is now labeled, paired with the logical form that corresponds to the detected canonical utterance paraphrase c_x.

We note that an alternative modeling choice was to use the semantic parser itself as the scorer for candidate canonical utterances. However, decoupling the parser and the scorer is beneficial, as the discriminative scoring function $s_t(\cdot)$ benefits from negative examples (incorrect paraphrases), unlike the generative semantic parser.

The success of our procedure depends on a good initial scoring function $s_0(x, c)$, to be used in the first iteration, that we next describe.

Initial Scoring Function We implement $s_0(x, c)$ in an unsupervised manner, as no labeled examples are available in the first iteration. Formally, we take $s_0(x, c) = -WMD(x, c)$, where $WMD(x, c)$ is the Word Mover's Distance (WMD) (Kusner et al., 2015) between x and c, which is the minimum amount of distance that the embedded words of one utterance need to travel to reach those of the other utterance. We found WMD to perform better than cosine similarity over averaged pre-trained embeddings, as WMD performs word-level alignment, and shared words (such as entities) encourage small distance.

Implementation details We take the unlabeled utterance set to be all utterances in $\mathcal{D}_{\text{nat}}$, when ignoring logical forms: $\mathcal{X}_{\text{ul}} = \{x \mid (x, \cdot) \in \mathcal{D}_{\text{nat}}\}$. We generate all canonical utterances up to depth D= 6, resulting in roughly 350K canonical utterances in SCHOLAR and GEOQUERY, and coverage of 90% of the examples in $\mathcal{X}_{\text{ul}}$ (Table 2).

Our binary classifier $s_t(\cdot)$ is trained from paraphrases detected by workers. We utilize the ESIM neural architecture originally used for natural language inference (Chen et al., 2017). We also employ ELMo contextualized embeddings (Peters et al., 2018), and minimize the binary cross-entropy loss for each example.

5 Experiments

5.1 Experimental Setup

Datasets We experiment with two popular semantic parsing datasets: GEOQUERY (Zettlemoyer and Collins, 2005) and SCHOLAR (Iyer et al., 2017), that contain questions about US geography and academic publications, respectively.

Because we utilize the original grammar from Wang et al. (2015) that generates logical forms in lambda-DCS (Liang, 2013), we first manually annotated GEOQUERY and SCHOLAR with lambda-DCS logical forms that are translations of the original logical forms (Prolog for GEOQUERY and SQL for SCHOLAR). We only convert examples that are covered by the OVERNIGHT grammar, which results in annotating 99.3% of the examples in GEOQUERY (874 in total), and 96.7% of the examples in SCHOLAR (790 in total).

Grammar Generation To generate the training data $\mathcal{D}_{\text{on}}$ for our OVERNIGHT baseline and GRANNO, we first exhaustively generated logical form and canonical utterance pairs from the grammar up to depth D $= 5$ for OVERNIGHT and D $= 6$ for GRANNO, using type matching rules such that vacuous logical forms are not generated. Then, we further pruned unlikely logical forms that can be automatically detected (e.g., contradictions such as *"state that borders california and that not borders california"*). We additionally pruned equivalent examples: if we generated a logical form with the structure `A ⊓ B`, we pruned `B ⊓ A`.

Crowd Sourcing We gathered annotations from crowd workers by running a qualification task where we manually verified workers (annotators with at least 85% success rate were qualified). Then, qualified workers performed the tasks for the remaining examples, and for those where unqualified workers failed.

For OVERNIGHT, we gathered a single paraphrase per canonical utterance in $\mathcal{D}_{\text{on}}$ with a cost of 0.06$. For GRANNO, we detected a single paraphrase for each unlabeled utterance in $\mathcal{X}_{\text{ul}}$ that was not annotated in previous iterations, with a cost of 0.05$.

In total, we gathered 7,140 paraphrases for OVERNIGHT with a total cost of 515$, and 2,594 detections for GRANNO with a total cost of 155$. Thus, GRANNO had lower cost per task, and given the results below, benefits more from fewer tasks.

Neural Semantic Parser We use a standard semantic parser provided by AllenNLP (Gardner et al., 2017), based on a sequence-to-sequence model (Sutskever et al., 2014). The encoder is a BiLSTM that receives ELMo pre-trained embeddings (Peters et al., 2018) as input. The attention-based decoder (Bahdanau et al., 2015) is an LSTM that also employs copying (Jia and Liang, 2016).

Model	GEOQUERY	SCHOLAR
SUPERVISED	86.3	83.4
GRANNO	72.0	69.2
OVERNIGHT	61.9	40.8
OVERNIGHT-ORACLE-LF	77.7	73.5
GRANNO-ORCALE	79.1	72.5

Table 5: Denotation accuracy on the test set.

Training Scheme For both the semantic parser and the paraphrase detection model $s_t(\cdot)$ we take 10% of the training data for validation (or the official development set, if it exists). When training both models, we abstract examples to their templates, as in Dong and Lapata (2016) and Finegan-Dollak et al. (2018) inter alia. For our semantic parser, we tune the learning rate and dropout over the development set, and for $s_t(\cdot)$ we use the same hyper-parameter values as in Chen et al. (2017). We use early-stopping, and choose the model with the highest denotation accuracy and F_1 measure for the semantic parser and $s_t(\cdot)$, respectively.

5.2 Results

Main Results Table 5 shows the denotation accuracy for all experiments, when training from: $\mathcal{D}_{\text{nat}}$ (SUPERVISED); $\mathcal{D}_{\text{on}}$ (OVERNIGHT); $\mathcal{D}_{\text{lang}}$, as described in Section 3.2 (OVERNIGHT-ORACLE-LF); $\mathcal{D}_{\text{ga}}$, described in Section 4 (GRANNO); and $\mathcal{D}_{\text{ga}}$ where we simulate a perfect worker that always detects the gold paraphrase if it appears in the top-K candidates (GRANNO-ORCALE).

Results show that OVERNIGHT achieves substantially lower accuracy compared to training with examples from the target distribution (SUPERVISED), inline with the analyses presented in Section 3. For instance, SCHOLAR accuracy more than halves (40.8 for OVERNIGHT in comparison to 83.4 for SUPERVISED). Conversely, our suggested method GRANNO that directly annotates unlabeled utterances achieves much higher accuracy than OVERNIGHT. Utilizing crowd workers for detection leads to 70.6 accuracy on average, and to 75.8 when simulating perfect crowd workers (GRANNO-ORACLE).

All models performed well on the development set that was sampled from the same distribution as the training set ($> 80\%$ accuracy), and thus differences in performance are due to generalization to the true distribution. Moreover, the accuracy of the paraphrase detection model $s_t(\cdot)$ was very high ($> 95\%$ F_1 measure) on the develop-

Case	%	Natural language utterance	Gold canonical utterance	Detected canonical utterance
us	.48	*"what is the highest point in california?"*	*"place that is high point of california"*	*"high point of california"*
os	.13	*"how many citizens in california?"*	*"population of california"*	*"total population of california"*
w	.08	*"which state has the greatest density?"*	*"state that has the largest density"*	*"total density of state"*
pw	.31	*"what is the lowest elevation in california?"*	*"place that is low point of california"*	*"elevation of low point of california"*

Table 6: Examples of false positive detections by crowd workers for different cases: *under specification* (*us*), *over specification* (*os*), *wrong* (*w*) and *partially wrong* (*pw*).

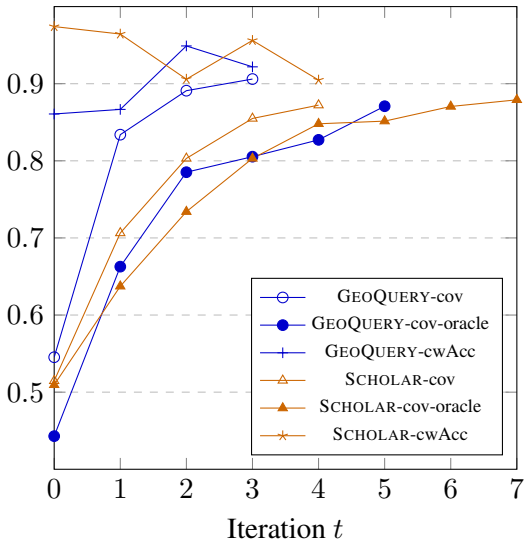

Figure 4: Analysis of GRANNO across iterations.

ment set, showing that detection is easier to model compared to generation.

We note in passing that we also implemented a baseline that uses unlabeled examples in conjunction with OVERNIGHT through self-training (Konstas et al., 2017). We trained a model with OVERNIGHT and iteratively labeled unlabeled utterances for which model confidence was high. However, we were unable to obtain good performance with this method.

GRANNO Analysis GRANNO iteratively annotates unlabeled utterances utilizing crowd workers. Figure 4 reports several metrics for GRANNO for each iteration: *cov* details the fraction of annotated utterances in $\mathcal{X}_{\text{ul}}$, where *cov-oracle* corresponds to the annotation coverage by GRANNO-ORACLE. In addition, cwAcc is the crowd workers' detection accuracy per iteration, with respect to the gold canonical utterances. The figure shows that in both datasets, GRANNO converges in a few iterations, and that workers' detection accuracy is high across all iterations ($> 85\%$).

An interesting phenomenon is that crowd workers (GRANNO) cover the unlabeled utterances faster than oracle workers (GRANNO-ORACLE). To analyze this, we inspect false positive, i.e., cases where the gold canonical utterance does *not* appear in the top-K candidates of $s_t(\cdot)$, but the crowd worker detects some candidate as the paraphrase. Table 6 presents examples for these cases and their fraction within all false positives for iteration $t = 0$, where workers cover unlabeled utterances faster than the oracle. We find that in 61% of the cases, the choice of the workers was equivalent to the gold candidate. This is due to *under specification*, when the gold paraphrase is more specific than the detected one, or *over specification*, which is the opposite case. The other 39% are indeed errors, which we split into *wrong* detections, and *partially wrong* detections, where the detected paraphrase is different than the gold one, but is reasonable choice the phrasing of the question. E.g., for *"what is the lowest elevation in California"* it is unclear whether the answer should be a location or the elevation. This shows that many false positives are in fact correct.

Limitations GRANNO relies on the unsupervised function $s_0(\cdot)$ to bootstrap the annotation procedure. In both datasets, $s_0(\cdot)$ managed to rank gold paraphrases within its top-5 candidates for roughly half of the unlabeled utterances in $\mathcal{X}_{\text{ul}}$, but this is not guaranteed.

During each iteration in GRANNO, the function $s_t(\cdot)$ is applied on all pairs of an unlabeled utterance and candidate canonical utterances, thus $s_t(\cdot)$ is applied $O(|\mathcal{X}_{\text{ul}}| \cdot |\mathcal{D}_{\text{on}}|)$ times. Empirically, we find that computation time is manageable when limiting the application of $s_t(\cdot)$ to candidates that share the same entities as the unlabeled utterance. However, this might not suffice for KBs with large schemas. In such cases, an information retrieval module could retrieve a small number of candidates, similar to Yang et al. (2019).

6 Related Work

Several works used a human in the loop for training a semantic parser. Iyer et al. (2017) incorporated user feedback to detect wrong parses and

sent them to expert annotation. Lawrence and Riezler (2018) and Berant et al. (2019) improved a supervised parser by showing its predictions to users and training from this feedback. Gupta et al. (2018) built a hierarchical annotation scheme for annotating utterances with multiple intents. Labutov et al. (2019) trained a semantic parser through interactive dialogues. Comparing to these works, our proposed method requires no supervised data or expert annotators, and is suitable for rapid development of parsers in multiple domains.

In semi-supervised learning, Konstas et al. (2017) used self-training to improve an existing AMR parser. Others used a variational autoencoder by modeling unlabeled utterances (Kociský et al., 2016) or logical forms (Yin et al., 2018) as latent variables. However, empirical gains from the unlabeled data were relatively small compared to annotating more examples.

Finally, several papers extended the OVERNIGHT procedure. Ravichander et al. (2017) replaced phrases in the lexicon with images to elicit more natural language from workers. (Cheng et al., 2018) generated more complex compositional structures by splitting the canonical utterances into multiple steps. Such work relies on workers to *generate* paraphrases, while we propose to simply *detect* them.

7 Conclusion

We address the challenge of generating data for training semantic parsers from scratch in multiple domains. We thoroughly analyze the OVERNIGHT procedure, and shed light on the factors that lead to poor generalization, namely logical form mismatch and language mismatch. We then propose GRANNO, a method that directly annotates unlabeled utterances with their logical form, by letting crowd workers detect automatically-generated canonical utterances. We demonstrate our method's success on two popular datasets, and find it substantially improves generalization to real data, compared to OVERNIGHT.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was completed in partial fulfillment for the PhD degree of the first author, which was also supported by a Google PhD fellowship. This research was partially supported by The Israel Science Foundation grant 942/16, The Blavatnik Computer Science Research Fund and The Yandex Initiative for Machine Learning.

References

Y. Artzi and L. Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics (TACL)*, 1:49–62.

D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Jonathan Berant, Daniel Deutch, Amir Globerson, Tova Milo, and Tomer Wolfson. 2019. Explaining queries over web tables to non-experts. *International Conference on Data Engineering (ICDE)*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Jianpeng Cheng, Siva Reddy, and Mirella Lapata. 2018. Building a neural semantic parser from a domain ontology. *arXiv preprint arXiv:1812.10037*.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL)*.

L. Dong and M. Lapata. 2016. Language to logical form with neural attention. In *Association for Computational Linguistics (ACL)*.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-sql evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

J. Herzig and J. Berant. 2017. Neural semantic parsing over multiple knowledge-bases. In *Association for Computational Linguistics (ACL)*.

J. Herzig and J. Berant. 2018. Decoupling structure and lexicon for zero-shot semantic parsing. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: a new dataset for compositional question answering over real-world images. *arXiv preprint arXiv:1902.09506*.

S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, and L. Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Association for Computational Linguistics (ACL)*.

R. Jia and P. Liang. 2016. Data recombination for neural semantic parsing. In *Association for Computational Linguistics (ACL)*.

J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR)*.

T. Kociský, G. Melis, E. Grefenstette, C. Dyer, W. Ling, P. Blunsom, and K. M. Hermann. 2016. Semantic parsing with semi-supervised sequential autoencoders. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1078–1087.

I. Konstas, S. Iyer, M. Yatskar, Y. Choi, and L. Zettlemoyer. 2017. Neural AMR: sequence-to-sequence models for parsing and generation. *CoRR*, 0.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

T. Kwiatkowski, E. Choi, Y. Artzi, and L. Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Igor Labutov, Bishan Yang, and Tom Mitchell. 2019. Learning to learn semantic parsers from natural language supervision. *arXiv preprint arXiv:1902.08373*.

Carolin Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1820–1830, Melbourne, Australia. Association for Computational Linguistics.

P. Liang. 2013. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*.

P. Liang, M. I. Jordan, and D. Klein. 2011. Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)*, pages 590–599.

Nicholas Locascio, Karthik Narasimhan, Eduardo De Leon, Nate Kushman, and Regina Barzilay. 2016. Neural generation of regular expressions from natural language with minimal domain knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1918–1923, Austin, Texas. Association for Computational Linguistics.

J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *North American Association for Computational Linguistics (NAACL)*.

Abhilasha Ravichander, Thomas Manzini, Matthias Grabmair, Graham Neubig, Jonathan Francis, and Eric Nyberg. 2017. How would you say it? eliciting lexically diverse dialogue for supervised semantic parsing. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 374–383, Saarbrücken, Germany. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Y. Su and X. Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Empirical Methods in Natural Language Processing (EMNLP)*.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3104–3112.

Y. Wang, J. Berant, and P. Liang. 2015. Building a semantic parser overnight. In *Association for Computational Linguistics (ACL)*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. *arXiv preprint arXiv:1902.01718*.

Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. StructVAE: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–765, Melbourne, Australia. Association for Computational Linguistics.

M. Zelle and R. J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1050–1055.

L. S. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Uncertainty in Artificial Intelligence (UAI)*, pages 658–666.

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