The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

Merging Weak and Active Supervision for Semantic Parsing

Ansong Ni, Pengcheng Yin, Graham Neubig
Carnegie Mellon University
{ansongn, pcyin, gneubig}@cs.cmu.edu

Abstract

A semantic parser maps natural language commands (NLs) from the users to executable meaning representations (MRs), which are later executed in certain environment to obtain user-desired results. The fully-supervised training of such parser requires NL/MR pairs, annotated by domain experts, which makes them expensive to collect. However, weakly-supervised semantic parsers are learnt only from pairs of NL and expected execution results, leaving the MRs latent. While weak supervision is cheaper to acquire, learning from this input poses difficulties. It demands that parsers search a large space with a very weak learning signal and it is hard to avoid spurious MRs that achieve the correct answer in the wrong way. These factors lead to a performance gap between parsers trained in weakly- and fully-supervised setting. To bridge this gap, we examine the intersection between weak supervision and active learning, which allows the learner to actively select examples and query for manual annotations as extra supervision to improve the model trained under weak supervision. We study different active learning heuristics for selecting examples to query, and various forms of extra supervision for such queries. We evaluate the effectiveness of our method on two different datasets. Experiments on the WikiSQL show that by annotating only 1.8% of examples, we improve over a state-of-the-art weakly-supervised baseline by 6.4%, achieving an accuracy of 79.0%, which is only 1.3% away from the model trained with full supervision. Experiments on WikiTableQuestions with human annotators show that our method can improve the performance with only 100 active queries, especially for weakly-supervised parsers learnt from a cold start.[1]

Introduction

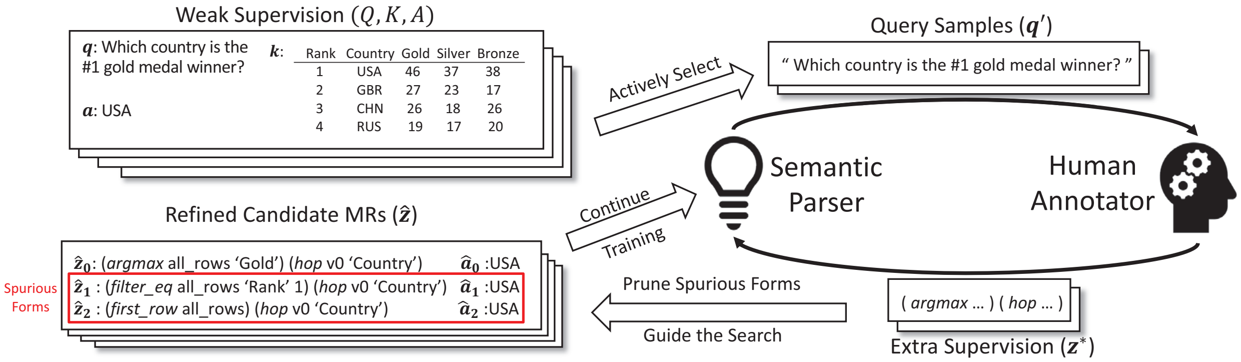
Semantic parsing maps a user-issued natural language (NL) utterance to a machine-executable meaning representation (MR), such as $\lambda-$calculus (Zettlemoyer and Collins 2005), SQL queries (Dong and Lapata 2018) or general purpose programming languages (*e.g.*, Python) (Yin and Neubig 2018). These MRs can then be executed in a certain environment (*e.g.*, a knowledge base, KB) to achieve the goal of users, such as querying a KB using natural language.

Classical supervised learning of semantic parsers requires parallel corpora of NL utterances and their corresponding annotated MRs (Zettlemoyer and Collins 2005; Dong and Lapata 2018; Rabinovich, Stern, and Klein 2017). However, this annotation requires strong domain expertise in the schema of the MR language (*e.g.*, SQL of KB queries), and hiring such domain experts can be expensive. Moreover, depending on the type of meaning representation used, the same input utterance could be grounded to multiple, equally acceptable MRs with diverse syntactic form (*e.g.* different ways to implement a loop), making it non-trivial to validate the correctness of annotated MRs given by different annotators. These factors make it very expensive to create fully-supervised semantic parsing datasets at large scale.

Weakly-supervised semantic parsing aims to solve this problem by relaxing the requirement for annotated MRs, instead training the parser using indirect supervision of *only* the expected result of executing an MR corresponding to user's intent (*i.e.*, an answer for a user question) (Clarke et al. 2010; Berant et al. 2013; Pasupat and Liang 2015). Compared with annotation-intensive MRs, such expected results are easy to obtain, even from annotators who do not know the specific MR language. Still, weakly supervised learning of semantic parsers remains a non-trivial task. First, the search space of possible MRs given an execution result is exponentially large, while the reward the parser receives during training is sparse and binary (*e.g.* one *iff* the execution result is correct); Another issue is *spurious MRs*, which are programs that happens to execute to the correct result, but are semantically incorrect. For example, a parser may wrongly parse the utterance "multiply two by two" to "2+2" and still get the correct answer. Such spuriousness would add a great amount of noise to the already-weak supervision signal.

To tackle these challenges with weakly-supervised semantic parsing, we propose a method to combine Weak and Active Supervision for Semantic Parsing (WASSP). As illustrated in Fig. Fig. 1, for each iteration, we first train the semantic parsing model to convergence with weak supervision. Then we perform active learning which allows the model to actively ask for extra supervision (*e.g.* annotated MRs) for only the small fraction of examples that would

[1] Code available at https://github.com/niansong1996/wassp


Weak Supervision (Q, K, A)
$\boldsymbol{q}$: Which country is the #1 gold medal winner?
$\boldsymbol{k}$:
$\boldsymbol{a}$: USA
Query Samples $(\boldsymbol{q}')$
" Which country is the #1 gold medal winner? "
Actively Select
Semantic Parser
Human Annotator
Refined Candidate MRs $(\hat{\boldsymbol{z}})$
Continue Training
$\hat{\boldsymbol{z}}_0$: (argmax all_rows 'Gold') (hop v0 'Country') $\hat{\boldsymbol{a}}_0$:USA
Spurious Forms
$\hat{\boldsymbol{z}}_1$: (filter_eq all_rows 'Rank' 1) (hop v0 'Country') $\hat{\boldsymbol{a}}_1$:USA
$\hat{\boldsymbol{z}}_2$: (first_row all_rows) (hop v0 'Country') $\hat{\boldsymbol{a}}_2$:USA
Prune Spurious Forms
Guide the Search
(argmax ...) (hop ...)
Extra Supervision $(\boldsymbol{z}^*)$

Rank	Country	Gold	Silver	Bronze
1	USA	46	37	38
2	GBR	27	23	17
3	CHN	26	18	26
4	RUS	19	17	20

Figure 1: An overview of our proposed WASSP: we first train a semantic parser with weak supervision till convergence, then a small fraction of NLs are selected from the training set to actively query the annotator for extra supervision, which is then used to refined the candidate MRs for the next round of weakly-supervised training.

maximally improve its performance. Finally, we feed the extra supervision to the semantic parsing model to rule out spurious logical forms and guide the search for hard examples.

Given our proposed WASSP framework, we examine three main research questions:

RQ1 How effective is WASSP in improving the performance of weakly-supervised semantic parser?

RQ2 How can we pick the set of utterances where direct supervision will maximally benefit parsing accuracy?

RQ3 What kind of extra supervision can WASSP use to improve the model for each sampled utterance?

Specifically, to answer RQ1, we measure the improvement of WASSP over a weakly-supervised baseline with varying number of queries made. To answer RQ2, we investigate different heuristics to select a smallest set of NL utterances for annotation, by considering the *informativeness*, *representativeness* and *diversity* of an example. To answer RQ3, we study various forms of extra supervision to provide to the model for each sampled utterance. While just presenting the corresponding fully-specified MRs would be a trivial strategy, we show that this is not our only choice. Instead, using a simpler form, the *sketch* of an MR, the actively-supervised semantic parsers can achieve similar performance compared to using the resource-heavy MRs.

We evaluate the effectiveness of our proposed active learning approach on two different datasets: the WikiSQL dataset (Zhong, Xiong, and Socher 2017) and the WikiTableQuestions dataset (Pasupat and Liang 2015). We demonstrate that active learning can greatly improve the performance semantic parsers trained with weak supervision with only a small fraction of examples actively selected and queried: By querying a total fraction of 1.8% of the training examples for WikiSQL, we improve over a state-of-the-art weakly-supervised baseline by 6.4%, achieving an accuracy of 79.0%, which is only 1.3% away from the model trained with full supervision. Experiment results on WikiTableQuestions show that WASSP can boost the performance of a model suffered from a cold start by 25.6% with only 50 examples. We also show that alternative forms of extra supervision to fully-specified MRs as MR sketches may also be used by WASSP and yields similar performances, only lose an maximum of 1.4% of accuracy comparing to full gold MR annotations on the WikiSQL dataset.

Weakly Supervised Semantic Parsing

Formulation

Formally, given a natural language utterance $\boldsymbol{q}$, a semantic parser is a parametric model $P(\boldsymbol{z}|\boldsymbol{q};\theta)$ that transduces the utterance $\boldsymbol{q}$ into a meaning representation $\boldsymbol{z}$. The parser could be optimized using either classical supervised learning, or weakly supervised learning.

In the classical setting of (fully-) supervised learning, a model is optimized using a parallel corpus D consisting of paired NL utterances and their annotated MRs, *i.e.*, $D = \{\langle \boldsymbol{q}_i, \boldsymbol{z}_i \rangle\}_{i=1}^{N}$. The standard maximum likelihood learning problem is:

$$\operatorname*{argmax}_{\theta} \prod_{\langle \boldsymbol{q}_i, \boldsymbol{z}_i \rangle \in D} P(\boldsymbol{z}_i|\boldsymbol{q}_i;\theta). \tag{1}$$

Weakly-supervised learning can be formulated as a reinforcement learning problem, instead of access to gold-standard MRs, the semantic parser (*i.e.*, the agent) is only presented with an execution model $\boldsymbol{k}$ (*i.e.*, the environment, such as a database) and the gold execution result $\boldsymbol{a}$. The parser interacts with $\boldsymbol{k}$, searching for high-reward MRs $\hat{\boldsymbol{z}}$ that execute to the correct results (*i.e.*, $\boldsymbol{k}(\hat{\boldsymbol{z}}) = \boldsymbol{a}$). A training instance is therefore defined by a triplet: utterance $\boldsymbol{q}$, execution environment $\boldsymbol{k}$ and gold answer $\boldsymbol{a}$. The learning objective is to maximize the probability of the correct answer $\boldsymbol{a}$ by marginalizing over all candidate MRs that execute to $\boldsymbol{a}$:

$$\begin{aligned} &\operatorname*{argmax}_{\theta} \prod_{\langle \boldsymbol{q}_i, \boldsymbol{z}_i \rangle \in D} P(\boldsymbol{a}_i|\boldsymbol{q}_i;\theta) \\ =&\operatorname*{argmax}_{\theta} \prod_{\langle \boldsymbol{q}_i, \boldsymbol{z}_i \rangle \in D} \sum_{\{\hat{\boldsymbol{z}}_i \in \hat{Z} | \boldsymbol{k}(\hat{\boldsymbol{z}}_i) = \boldsymbol{a}_i\}} P(\hat{\boldsymbol{z}}_i|\boldsymbol{q}_i;\theta) \end{aligned} \tag{2}$$

Weakly supervised parsers are usually trained with an EM-like method: the model searches for high-reward MRs by sampling from the semantic parser, then the model parameters are optimized to maximize the probability of generating these high-reward MRs.

Difficulties with Weak Supervision

Two challenges make weakly-supervised parsing harder than its fully-supervised counterpart:

Exploration of an Exponentially-large Space. The search space of latent MRs is exponential or infinite, making it intractable to exhaustively search over the set of plausible MRs $\{\hat{\boldsymbol{z}}_i | \boldsymbol{k}(\hat{\boldsymbol{z}}_i) = \boldsymbol{a}_i\}$ to calculate sum in Eq. (2). Thus, it is common to resort to sampling or k-best search to approximate this space (Guu et al. 2017), potentially combined with sophisticated methods to reduce the space of plausible MRs with type constraints (Krishnamurthy, Dasigi, and Gardner 2017) or use of a memory buffer to cache high-reward explored MRs (Liang et al. 2018). However, inferring MRs for complex, highly compositional input utterances still remains an open challenge (Pasupat and Liang 2016).

Spurious MRs. In contrast with the full-supervised setting where the parser is optimized using the semantically correct gold-standard MR for each utterance, the search space of latent MRs contains spurious samples that happen to execute to the correct results despite being semantically incorrect. For example, in Fig. 1, the NL utterance *"Which country is the #1 gold medal winner?"* has one semantically correct MR z_0 while MRs z_1 and z_2 do not match the semantics of the original utterance. Nonetheless, z_1 and z_2 follow the grammatical rules, satisfy the type constraints, and get correct results, and thus there is no simple way for a weakly supervised learner to distinguish them from gold-standard MRs. Such spurious MRs could be exponentially many (Pasupat and Liang 2016), and add significant noise to the training process. While attempts have been made to mitigate the issue of spuriousness, such as using a ranker trained with extra semantic or lexicon information (Cheng and Lapata 2018) or introduce prior knowledge to bias the policy (Misra et al. 2018), it is still highly non-trivial to totally solve this issue due to the ambiguous nature of the reward.

Empirically, these challenges lead to performance gap between fully- and weakly-supervised semantic parsers. For example, on the WikiSQL dataset, the state-of-the-art fully-supervised semantic parser reaches an testing execution accuracy of 86.2% (Hwang et al. 2019) while the best weakly-supervised semantic parser (Liang et al. 2018) only has an accuracy of 72.6%, a difference of 13.6%.

Merging Weak and Active Supervision

In order to solve these two problems, we first discuss three key insights about weakly-supervised semantic parsing that motivates WASSP:

1) Though mixed with spurious MRs, weakly-supervised semantic parsers are still able to find gold MRs for simpler input utterances, even with such noisy and weak learning signal. This means that effective training of semantic parsers does not require full supervision using annotated MRs for *all* examples.

2) The EM-like weakly-supervised optimization will "stall" at the point when no new high-reward MRs can be found by the parser, and the set of discovered MRs with correct answers are unchanged for all examples in the training set. Eq. (2) will then converge to a local optimum. However, if the parser can discover new MRs with correct execution results, growing the set of high-reward MRs for even a small fraction of training examples, the optimization of Eq. (2) will resume, and the parser could in turn explore more similar high-reward MRs with its updated parameters. In other words, it is possible that some extra supervision for only a small fraction of the training data can prevent stagnation and resume the optimization process.

3) Fully-specified MRs might not always be necessary to provide extra supervision for a weakly-supervised parser to discover high-reward MRs for complex utterances or rule out spurious MRs for ambiguous inputs. As an example, to identify the semantically correct MR for the illustrative example in Fig. 1, the parser only need to be informed that the gold MR should contain a superlative operation (*i.e.*, `argmax`). This motivates us to explore alternative forms of extra supervision other than fully-specified MRs.

Based on these three insights, we propose WASSP, a framework to merge active and weak supervision for semantic parsing. Fig. 1 presents a schematic overview. Specifically, WASSP iteratively performs the following steps:

Step 1: Train the semantic parser $P(\hat{\boldsymbol{z}}|\boldsymbol{q};\theta)$ to convergence by optimizing Eq. (2) over the training set D and discovered gold MR candidates $\hat{Z}$;

Step 2: With active sample selection heuristic, select a small subset of the training set $D' \subseteq D$ within certain budget and query the annotator for extra supervision for examples in D';

Step 3: After receiving annotations, *e.g.*, in the form of annotated MRs $\{\boldsymbol{z}^*_j | (\boldsymbol{k}_j, \boldsymbol{q}_j, \boldsymbol{a}_j) \in D'\}$, update the set of high-reward MRs $\hat{Z}$ explored so far for each training instance in D'. If the form of extra supervision is annotated MRs, this simply amounts to set $\hat{Z}$ to only contain the annotated MR, *i.e.*, $\hat{Z} = \{\boldsymbol{z}^*\}$. We will introduce other form of extra supervision in later sections.

The motivation behind WASSP is very simple; every time weakly-supervised training stalls, WASSP allows the parser to select a small fraction of the training set and query for extra supervision, and use the received extra supervision to help continue the training process. In the following sections, we examine two design decisions: (1) how to select our subset of data to query D' , and (2) what varieties of supervision can WASSP utilize.

Query Sample Selection Heuristics

In this section, we examine strategies to select examples that are more likely to improve the performance of the semantic parser. First we propose a very simple yet effective *correctness-based* method, then we investigate two types of commonly adopted heuristics in active learning, *uncertainty-based* methods and *coverage-based* methods.

Correctness-based Method

In traditional *pool-based active learning* (Lewis and Gale 1994), the learner selects from a pool of examples without any information about the true label of example. However, for weakly-supervised semantic parsing we do have the expected execution result $\boldsymbol{a}$ as indirect information that can be used to inform our choice of data to annotate.

As noted above, an MR executing to the correct result does not necessarily entail that the final logical form is correct ($\boldsymbol{k}(\hat{\boldsymbol{z}}) = \boldsymbol{a} \nRightarrow \hat{\boldsymbol{z}} = \boldsymbol{z}$) because the MR may be spurious. However, if an MR does not execute to the correct result, this *does* entail that the MR is incorrect ($\boldsymbol{k}(\hat{\boldsymbol{z}}) \neq \boldsymbol{a} \Rightarrow \hat{\boldsymbol{z}} \neq \boldsymbol{z}$). Taking advantage of this fact, we can derive a *correctness-based heuristic*, which prefers to select the examples $\boldsymbol{q}$ for which there is no MR generated by the parser that matches the expected execution result thus no learning signal (*i.e.* reward) is provided for the semantic parser. These examples are much more likely to benefit from extra supervision than examples where the parser already has found at least one semantic parse corresponding to a correct answer.

Uncertainty-based Methods

Uncertainty-based methods are a standard set of active learning techniques (Lewis and Catlett 1994; Tong and Koller 2001; Shen et al. 2004) that attempt to maximize the *informativeness* of selected examples by preferring examples where the learner is most uncertain about its prediction. In the context of semantic parsing, we follow (Duong et al. 2018) and measure the uncertainty on an example $\boldsymbol{q}$ by the **least confidence score** (Culotta and McCallum 2005):

$$\boldsymbol{q}' = \arg\min_{\boldsymbol{q} \in D} [\max_{\tilde{\boldsymbol{z}} \in Z} P(\tilde{\boldsymbol{z}} | \boldsymbol{q}; \theta)] \qquad (3)$$

Though we can not enumerate all $\boldsymbol{z}$ in the MR space Z, it is easy to approximate this measurement by only enumerating all MRs remain in the beam Z_B after beam search.

Coverage-based Methods

Coverage-based methods are another common family of methods in active learning (Dasgupta and Hsu 2008). This type of methods consider the *representativeness* of the selected examples and attempts to cover as many other unselected examples as possible. Here we explore two different kinds of methods that adopt the idea of *representativeness*:

Failed Word Coverage: Some particular words may be particularly difficult for semantic parsers to recognize, either because they are infrequent, or because they are only loosely connected to the MR. Based on this fact, we introduce a heuristic to select the examples with the largest number of words that are more prone to cause failure. An example is labeled as `fail` if the MR given by the semantic parser can not execute to the expected result and we denote the set of `fail` examples in the training set as $D^- \subseteq D$. Then given a bag-of-words representation for example $\boldsymbol{q} = [q^1, ..., q^m]$ where m is the vocabulary size, we estimate the probability of a word q^j that leads to `fail` by counting the occurrences of this word in the `fail` examples and all examples:

$$P(\texttt{fail}|q^j) = \sum_{\boldsymbol{q} \in D^-} q^j / \sum_{\boldsymbol{q} \in D} q^j$$

Then we select the examples that covers more of words that are more likely to cause failure as following:

$$\boldsymbol{q}' = \arg\max_{\boldsymbol{q} \in D} \prod_{j=1}^{m} q^j \cdot P(\texttt{fail}|q^j)$$

NL Utterance $\boldsymbol{q}$: *Which country is the #1 gold medal winner?*
Full MR $\boldsymbol{z}_f$:

```
(argmax all_rows 'Gold')
(hop v0 'Country')
```

MR Sketch $\boldsymbol{z}_s$:

```
(argmax ...) (hop ...)
```

Figure 2: An example of fully-specified MR and MR sketch

Clustering: For this method, we attempt to cluster together similar NL utterances. For each example, a sentence-level embedding is computed by averaging the pre-trained GloVe embedding (Pennington, Socher, and Manning 2014) of the words, then we adopt the K-Means algorithm to perform clustering of the training examples. Given a clustering of the training examples, we first rank the clusters by their size and leave out the last 20% of the clusters to reduce the risk of selecting examples that are not representative (*i.e.* outliers). Then from each of the remaining clusters, we random sample equal number of examples to encourage *diversity*.

Forms of Extra Supervision

In addition to the trivial solution of using fully-specified MRs as extra supervision, here we introduce an example of other forms of annotations that WASSP can utilize by merging them with the weakly-supervised training: the MR sketch. An example of a fully-specified MR and its sketch are shown in Fig. 2.

Fully-specified MRs Using fully-specified MRs as extra supervision for the selected query examples is an obvious strategy. We define the fully-specified MRs as a sequence of complete operations or function calls with all the variables or arguments and ready to be executed. An example of this is shown in Fig. 2 as $\boldsymbol{z}_f$. A fully-specified MR can simply mount the explored high-reward MRs and be directly it used for training.

MR Sketches A *sketch* of an executable MR is the sequence of operators or function names that does not fill in variables or arguments. An example MR sketch is shown in Fig. 2 as $\boldsymbol{z}_s$. With MR sketch annotation, we can: 1) Remove spurious high-reward MRs where the sketch does not match the gold MR sketch from $\hat{Z}$; 2) Use this sketch as a guide for future exploration (*e.g.* constrained decoding), the high-reward MRs are only saved in $\hat{Z}$ if their MR sketches match the gold MR sketch. These sketches provide several benefits:

1) Reduction of search space. Though MR sketches do not completely remove ambiguity, with the sketch, the parser only needs to fill out variables, and thus the search space is also greatly reduced. Moreover, since most neural semantic parsing models adopt a copy mechanism, variable type matching, or syntactic constraints (Krishnamurthy, Dasigi, and Gardner 2017; Liang et al. 2016), the search space becomes even more restricted and leaves little room for spurious MRs making it much easier to explore. For example, all spurious forms are pruned when given the gold MR sketch of `(argmax ...)(hop ...)` in Fig. 1;

2) Increased generality over the MR. Compared to annotating with fully-specified MRs (as z_f in Fig. 2), a sketch

Dataset	WikiSQL	WikiTableQuestions
#Tables	26,531	2,108
#Questions	80,654	22,033
NL Question	✓	✓
Table Content	✓	✓
Gold MR	✓	✗
Gold Exec. Result	✓	✓

Table 1: Statistics of the datasets used in the experiments.

is a higher-level abstraction, which resembles procedural knowledge for the model to parse this type of NL utterances without being deeply coupled with the specific details of the example itself. Take the example in Fig. 2, after annotating this example with its MR sketch `(argmax ...)(hop ...)`, annotators can easily annotate other similar NL queries as "Which country has the largest population" or "Which team was ranked the last" without wasting time filling the detailed arguments. This special trait may speed up the manual annotation process thus could potentially be cheaper to obtain.

Experiments

Experiment Setup

Dataset: We evaluate the performance of WASSP on two different datasets: WikiSQL (Zhong, Xiong, and Socher 2017) and WikiTableQuestions (Pasupat and Liang 2015). The statistics of these two datasets are shown in Table 1. The WikiSQL dataset provides both gold MRs and their expected execution results, making it possible to perform simulated experiments where we train the same model with weak, active, or full supervision. WikiTableQuestions does not have annotated MRs, and thus we perform only limited active learning experiments querying real human annotators to add additional MRs.

Neural Semantic Parsing Model: For the underlying neural semantic parsing model for WASSP, we adopt neural symbolic machines (NSM) (Liang et al. 2016), a strong weakly-supervised semantic parser. NSM uses a sequence-to-sequence network to transduce an input utterance into an MR. The recurrent decoder is augmented with a memory component to cache intermediate execution results in a partially generated MR (*e.g.*, the result of `argmax` for $\boldsymbol{z}_0$ in Fig. 1 is cached as v_0), which can be referenced by following operations (*e.g.*, in the `hop` function in $\boldsymbol{z}_0$). We follow (Liang et al. 2018), and train NSM using memory-augmented policy optimization (MAPO), which uses a memory buffer to cache currently explored high-reward MRs.

The NSM trained with MAPO is the state-of-the-art model for weakly-supervised semantic parsing on the WikiSQL dataset (detailed below); a single model (*i.e.* without ensemble) can reach an execution accuracy of 72.4% and 72.6% on the dev and test set, respectively. Upon inspection, we found that the syntax rules of NSM did not have full coverage of the SQL queries in the WikiSQL dataset, and we further augmented the coverage of the NSM syntax rules, improving this to 75.2% dev accuracy and 75.6% test accuracy with pure weak supervision. On the WikiTableQuestions dataset, an ensemble of 10 such models achieves the state-of-the-art performance of 46.3% execution accuracy while the performance of a single model is 43.1%(±0.5%). We use this as a strong baseline which we aim to improve.

Query Budget	Acc. (Imp.)
0 (Pure Weak Supervision)	75.6
100 (0.2%)	76.3(+0.7)
200 (0.4%)	76.7(+1.1)
500 (0.9%)	77.7(+2.1)
1,000 (1.8%)	78.6(+3.0)
2,500 (4.4%)	79.0(+3.4)
10,000 (17.7%)	79.8(+4.2)
56,355(All)	80.3
Previous Weak Supervision Methods	Acc.
NSM+MML (Liang et al. 2016)	70.7
NSM+MAPO (Liang et al. 2016; 2018)	72.6
Previous Full Supervision Methods	Acc.
STAMP (Sun et al. 2018)	74.4
TranX+AP (Yin and Neubig 2018)	78.6
Coarse2Fine (Dong and Lapata 2018)	78.5
TypeSQL+TC (Yu et al. 2018)	82.6

Table 2: WASSP with varying budget and previous fully- or weakly-supervised methods on WikiSQL. All methods use table contents during training and are evaluated on the test set.

Training Procedure: First we follow the procedure of (Liang et al. 2018) to train NSM with MAPO on both WikiSQL and WikiTableQuestions datasets with the same set of hyperparameters as used in the original paper. Then for WikiSQL, we run WASSP for 3 iterations with query budget 1,000 or more and only run for one iteration with smaller budget. For each iteration, the model queries for extra supervision and then it is trained for another 5K steps. The query budget is evenly distributed to these 3 iterations and limited by the total amount. For WikiTableQuestions, we simply run one such iteration (due to limit number of annotations obtained) but let it train for 50K steps with human annotated MRs.

Evaluation Metric: As with previous works (Pasupat and Liang 2015; Zhong, Xiong, and Socher 2017; Dong and Lapata 2018; Liang et al. 2018), we measure the execution *accuracy*, defined as the fraction of examples with correct execution results.

Main Results

Effectiveness of WASSP. First, to answer RQ1, we primarily investigate the performance of WASSP w.r.t. the amount of extra supervision during training. Table 2 and Table 3 list results of WASSP with varying query budgets on WikiSQL and WikiTableQuestions respectively. On the WikiTableQuestions dataset, MAPO has the option to begin with an empty memory buffer (*i.e.* a cold start) or warm up the memory buffer by searching with manually-designed prun-

Query Budget	Cold Start	Warm Start
0 (Pure Weak Supervision)	8.5	42.7
50 (0.5%)	34.1(+25.6)	42.7(+0.0)
100 (0.9%)	37.7(+29.2)	43.2(+0.5)

Previous Weak Supervision Methods	Acc.
(Pasupat and Liang 2015)	37.1
(Neelakantan et al. 2016)	34.2
(Haug, Ganea, and Grnarova 2018)	34.8
(Zhang, Pasupat, and Liang 2017)	43.7

Table 3: WASSP with varying budget and different types of weakly-supervised training start on WikiTableQuestions. Evaluated by execution accuracy on the test set, improvements are noted in brackets.

ing rules as heuristics (*i.e.* a warm start). Since designing the pruning heuristics also includes non-trivial human effort, we include the both results from a model trained from a cold start and a warm start to have a rather complete study. Our model presented here uses the correctness-based query selection heuristic and fully-specified MRs as extra supervision. We also compare WASSP with existing weakly- or fully-supervised methods.

From Table 2 we can see that with annotating only 100 (0.2%) examples from the training set, WASSP can improve the performance by 0.7% while querying 500 examples (0.89%) achieves an absolute improvement of 2.1% over the same model trained with pure weak supervision. Notably, if we allow a query budget of 1,000 examples, WASSP achieves an execution accuracy of 78.6%, a 6.0% absolute improvement over previous state-of-the-art result (72.6%) with only weak supervision (Liang et al. 2018). Additionally, with only 1.8% of annotated training examples, WASSP is already quite competitive against some of the strong fully-supervised systems trained with the whole annotated training set (Sun et al. 2018; Yin and Neubig 2018; Dong and Lapata 2018). This result is also only 1.7% away from the same semantic parser (NSM) trained with full supervision (80.3%). Finally, if we further increase the query budget to 10,000 examples, WASSP could achieve similar performance comparable with training the semantic parser using full supervision, but with 80% less annotated data. This set of results shows that WASSP allows learning with significantly less supervision.

We also have some interesting findings for the WikiTableQuestions dataset. While Table 3 shows that WASSP can improve the performance of a model trained with either cold or warm start, WASSP yields a substantial on models trained with a cold start: a 25.6% improvement with only 50 examples. Further investigations shows that the baseline model trained with a cold start on WikiTableQuestions is only able to correctly parse simple NL queries (*e.g.* count the number of rows, find maximum number, etc) that can be addressed with one statement (*e.g.* `(count ...)`, `(maximum ...)`). Since the model is under-trained, the probability of it exploring a high-reward complex MR for a sophisticated questions is very low. In this way, injecting a few annotations for the examples the model can not solve yet makes the model realize that more rewards may be obtained by more complex MRs, which in turn motivates and guides the exploration of the model.

Selection Heuristics	Dev.	Test
Baseline (No query made)	75.3	75.6
Random	76.2	76.4
Correctness	78.3	78.6
Uncertainty	76.4	77.0
+Correctness	**78.4**	**79.0**
Coverage-based (Fail Words)	77.1	77.4
Coverage-based (Clustering)	77.8	77.5

Table 4: Comparison of Different Query Sample Selection Heuristics. An equal query budget of 1,000 examples for gold MRs is given for all methods, performance measured by execution accuracy. Best numbers are in bold.

Query Sample Selection Heuristics. To answer RQ2, we evaluate WASSP trained with different sample selection heuristics introduced previously and summarize the results in Table 4. We found the simple correctness-based heuristic is effective, perhaps surprisingly so. Using uncertainty scores alone does not achieve good performance, barely exceeding random selection. Further investigation finds that by the end of weakly-supervised training, 84.0% of examples in training set with their parsed MRs not executed to the expected results have a confidence (Eq. (3)) over 0.9. This finding suggests that incorrect MRs could still have high model-assigned probability, which is worth future investigation.

Nevertheless, WASSP achieves the best performance by combining the two heuristics (*i.e.*, selecting the most uncertain examples among the failed ones)[2]. This query sample selection method combines the idea of correctness and uncertainty to maximizes the informativeness of the selected examples and reaches 78.4% and 79.0% of execution accuracy on dev. and test set. Finally, we found the coverage-based methods do not work as well, which may result from our relatively simple approach of measuring coverage using sentence embedding or failed words.[3]

Forms of Supervision Here we study how well WASSP can adapt to other forms of supervision besides the gold fully-specified MR. To answer RQ3, we conduct experiments with the MR sketch as extra supervision and compare its performance to using the fully-specified MR under the framework of WASSP and the results are shown in Table 5 and Table 6 for the WikiSQL and WikiTableQuestions datasets, respectively. In this set of experiments, we annotate examples selected with *correctness* heuristic.

From these tables, we can see that the performance of a model learnt from MR sketches as extra supervision is comparable to the model learnt from fully-specified MR,

[2] We also tried to combine other sets of heuristics but had equivalent or worse results than using one of the components.

[3] A stronger sentence embedding method may improve the performance, but a discussion of the best sentence embedding method is out of the scope of this paper.

Query Budget	Full MR	MR Sketch
1,000 (1.8%)	78.6	77.2 (-1.4)
2,500 (4.4%)	79.0	78.4 (-0.6)
10,000 (17.7%)	79.8	78.9 (-0.9)
56,355 (All)	80.3	79.1 (-1.2)

Table 5: Comparison of supervision forms for WikiSQL. Numbers in the brackets show the gap between full and sketched MRs.

Query Budget	Full MR	MR Sketch
50 (0.5%)	34.1	33.2 (-0.9)
100 (0.9%)	37.7	35.3 (-2.4)

Table 6: Comparison of Extra Supervision Forms for WikiTableQuestions. Numbers in the brackets show the gap between full and sketched MRs. Weakly-supervised training starts from a cold start.

regardless of the budget is larger (1,000 to 10,000, maximum gap of 1.4% on WikiSQL) or smaller (50 to 100, maximum gap of 2.4% on WikiTableQuestions). This shows that fully-specified MRs are not the only option as extra supervision for WASSP to achieve good improvement over weakly-supervised semantic parsers. We hope in future work to investigate other forms of extra supervision that may benefit WASSP.

Qualitative Analysis

Here we conduct a qualitative analysis on how the examples selected and queried by WASSP help it generalize to other unseen or not queried examples. Example 1 from Fig. 3 shows that extra supervision for the query set can generalize to other examples with similar NL utterance structure. For Example 2, the missing filter is corrected after receiving extra supervision for an example with similar conditions. As in Example 3, the parser can not use the correct filter for comparing time, but as the example above being selected and queried, the semantic parser trained with extra supervision learns to map “later than” to `filter_greater` for comparing time. These examples show that extra supervision for the actively queried examples, helps the model with WASSP learn to generalize to other similar test examples.

Related Work

(Artzi and Zettlemoyer 2013) proposed to jointly learn from meaning and context for. In the work of (Krishnamurthy and Mitchell 2012), they combined weak supervision from the knowledge base and the NL sentences. To handle the problem of spurious forms, (Muhlgay, Herzig, and Berant 2018) proposed a value-based search method with a trained critic network trained with the environment. In (Shen et al. 2004), they explored different active learning heuristics for named entity recognition, including informativeness, representativeness and diversity. (Duong et al. 2018) studied active learning for fully-supervised semantic parsing and showed that an uncertainty-based active learning method is powerful for traditional data collection but not useful to overnight data collection.

SELECTED-1 *How many visitors in 2007 were there?*

z_f
```
(filter_eq all_rows ‘2007’ ‘Year’)
(count v0)
```

z_s
```
(filter_eq ...) (count ...)
```

TESTED-1 *How many CIL competitions were there?*

$\hat{z}$
```
(filter_eq all_rows ‘CIL’ ‘Competition’)
(hop v0 ‘established’) ✗
```

$\hat{z}'$
```
(filter_eq all_rows ‘CIL’ ‘Competition’)
(count v0) ✓
```

SELECTED-2 *How many points did the driver who won $127,541 driving car #31 get?*

z_f
```
(filter_eq all_rows ‘$127,541’ ‘Winning’)
(filter_eq v0 ‘#31’ ‘Car’) (hop v1 ‘Pts’)
```

z_s
```
(filter_eq ...)(filter_eq ...)(hop ...)
```

TESTED-2 *Which driver won $40,000 in the NW Cup?*

$\hat{z}$
```
(filter_eq all_rows ‘NW Cup’ ‘Series’)
(hop v0 ‘driver’) ✗
```

$\hat{z}'$
```
(filter_eq all_rows ‘$40,000’ ‘Winning’)
(filter_eq all_rows ‘NW Cup’ ‘Series’)
(hop v0 ‘driver’) ✓
```

SELECTED-3 *What position was WIC in a year later than 2008?*

z_f
```
(filter_eq all_rows ‘WIC’ ‘Competition’)
(filter_greater v0 ‘2008’ ‘Year’)
(hop v1 ‘Position’)
```

z_s
```
(filter_eq ...)(filter_greater ...)
(hop ...)
```

TESTED-3 *What Chassis has a year later than 1989?*

$\hat{z}$
```
(filter_less all_rows ‘1989’ ‘Year’)
(hop v0 ‘Chassis’) ✗
```

$\hat{z}'$
```
(filter_greater all_rows ‘1989’ ‘Year’)
(hop v0 ‘Chassis’) ✓
```

Figure 3: Examples selected to query (SELECTED-) and similar test examples (TESTED-) with their top-1 parsed MR before ($\hat{z}$) and after ($\hat{z}'$) WASSP is applied. z_f and z_s denote full and sketched MR as extra supervision. Similar parts between query examples and test examples are underlined.

Conclusion

We propose WASSP, a framework to merge weak and active learning for semantic parsing. We study different query sample selection heuristics and various forms for extra supervision. Experiments on two datasets show that WASSP can greatly improve the performance of a weakly-supervised semantic parser with a small fraction of examples queried.

Acknowledgements

The authors thank Eduard Hovy, Alessandro Oltramari, and anonymous reviewers for their advice regarding this research. This work was supported in part by a gift from the Carnegie Bosch Institute.

References

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

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*.

Cheng, J., and Lapata, M. 2018. Weakly-supervised neural semantic parsing with a generative ranker. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 356–367.

Clarke, J.; Goldwasser, D.; Chang, M.; and Roth, D. 2010. Driving semantic parsing from the world's response. In *Proceedings of CoNLL*.

Culotta, A., and McCallum, A. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, 746–751.

Dasgupta, S., and Hsu, D. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, 208–215. ACM.

Dong, L., and Lapata, M. 2018. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*.

Duong, L.; Afshar, H.; Estival, D.; Pink, G.; Cohen, P.; and Johnson, M. 2018. Active learning for deep semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 43–48.

Guu, K.; Pasupat, P.; Liu, E. Z.; and Liang, P. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *arXiv preprint arXiv:1704.07926*.

Haug, T.; Ganea, O.-E.; and Grnarova, P. 2018. Neural multi-step reasoning for question answering on semi-structured tables. In *European Conference on Information Retrieval*, 611–617. Springer.

Hwang, W.; Yim, J.; Park, S.; and Seo, M. 2019. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*.

Krishnamurthy, J., and Mitchell, T. M. 2012. Weakly supervised training of semantic parsers. In *EMNLP-CoNLL*.

Krishnamurthy, J.; Dasigi, P.; and Gardner, M. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1516–1526.

Lewis, D. D., and Catlett, J. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*. Elsevier. 148–156.

Lewis, D. D., and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*, 3–12. Springer.

Liang, C.; Berant, J.; Le, Q.; Forbus, K. D.; and Lao, N. 2016. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *arXiv preprint arXiv:1611.00020*.

Liang, C.; Norouzi, M.; Berant, J.; Le, Q.; and Lao, N. 2018. Memory augmented policy optimization for program synthesis with generalization. *arXiv preprint arXiv:1807.02322*.

Misra, D.; Chang, M.-W.; He, X.; and Yih, W.-t. 2018. Policy shaping and generalized update equations for semantic parsing from denotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2442–2452.

Muhlgay, D.; Herzig, J.; and Berant, J. 2018. Value-based search in execution space for mapping instructions to programs. *arXiv preprint arXiv:1811.01090*.

Neelakantan, A.; Le, Q. V.; Abadi, M.; McCallum, A.; and Amodei, D. 2016. Learning a natural language interface with neural programmer. *arXiv preprint arXiv:1611.08945*.

Pasupat, P., and Liang, P. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of ACL*.

Pasupat, P., and Liang, P. 2016. Inferring logical forms from denotations. *arXiv preprint arXiv:1606.06900*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rabinovich, M.; Stern, M.; and Klein, D. 2017. Abstract syntax networks for code generation and semantic parsing. In *ACL*.

Shen, D.; Zhang, J.; Su, J.; Zhou, G.; and Tan, C.-L. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 589. Association for Computational Linguistics.

Sun, Y.; Tang, D.; Duan, N.; Ji, J.; Cao, G.; Feng, X.; Qin, B.; Liu, T.; and Zhou, M. 2018. Semantic parsing with syntax-and table-aware sql generation. In *ACL*.

Tong, S., and Koller, D. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2(Nov):45–66.

Yin, P., and Neubig, G. 2018. Tranx: A transition-based neural abstract syntax parser for semantic parsing and code generation. *arXiv preprint arXiv:1810.02720*.

Yu, T.; Li, Z.; Zhang, Z.; Zhang, R.; and Radev, D. 2018. Typesql: Knowledge-based type-aware neural text-to-sql generation. *arXiv preprint arXiv:1804.09769*.

Zettlemoyer, L. S., and Collins, M. 2005. Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 658–666. AUAI Press.

Zhang, Y.; Pasupat, P.; and Liang, P. 2017. Macro grammars and holistic triggering for efficient semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1214–1223.

Zhong, V.; Xiong, C.; and Socher, R. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.