# Overview of the NLPCC 2019 Shared Task: Open Domain Semantic Parsing

Nan DUAN

nanduan@microsoft.com

Microsoft Research Asia

**Abstract.** Semantic Parsing is a key problem for many artificial intelligence tasks, such as information retrieval, question answering and dialogue system. In this paper, we give the overview of the open domain semantic parsing shared task in NLPCC 2019. We first review existing semantic parsing datasets. Then, we describe open domain semantic parsing shared task in this year's NLPCC, especially focusing on the dataset construction. The evaluation results of submissions from participating teams are presented in the experimental part.

**Keywords:** Semantic Parsing, Natural Language Understanding, MSParS

## 1 Background

Semantic parsing aims to transform a natural language utterance into a machine executable meaning representation. It is one of the core technologies for building human-machine interaction engines, such as search, question answering and dialogue systems.

A number of semantic parsing datasets have been released in last decades, such as ATIS [1], JOBS [2], Geoquery [3], Free917 [4], WebQuestions [5], SimpleQuestions [6], and LC-QuAD [7]. However, these datasets are either limited by sizes and specific domains or biased on simple questions. ComplexWebQuestions [8] is a recently released semantic parsing dataset, which contains 34,689 questions with logical forms and focuses on 4 question types (multi-hop, multi-constraint, superlative and comparative). But as it uses WebQuestionsSP [9] as the seed for complex question generation, this dataset only covers 462 unique knowledge base (KB) predicates. WikiSQL [10] contains 80,654 <question, logical form> pairs, where each question is annotated based on one of 24,241 web tables. MSParS differs from WikiSQL in two ways: (i) MSParS is labeled based on a knowledge graph, while WikiSQL is labeled based on web tables. This leads to inference on MSParS posing a significant challenge as a knowledge graph is much more complicated than a single table; (ii) most questions in WikiSQL are multi-constraint ones, while MSParS contains more question types. In summary, the community lacks of a comprehensive semantic parsing dataset to evaluate semantic parsers from different perspectives.

## 2 Dataset Description

Motivated by the situation discussed above, we propose MSParS, a Multi-perspective Semantic ParSing dataset, for the NLPCC 2019 open domain semantic parsing shared task.

MSParS covers 9 types of single-turn questions: single-relation, CVT, multi-hop, multi-constraint, Yes/No, multi-choice, superlative, aggregation, and comparative. Considering the additional 3 types of multi-turn questions (multi-turn-entity, multi-turn-predicate, multi-turn-answer), the total number of the question types is 12. For each question, MSParS provides three kinds of annotations: the logical form of the question, the question type, and the parameters, i.e., the entities, types or values mentioned in the question while occurred in the logical form. Each logical form in MSParS is in form of untyped lambda-calculus and built based on a knowledge base and some predefined functions. We construct MSParS by crowd sourcing with pre-defined logical form patterns. Figure 1 shows those patterns and question examples of 9 single-turn types and 3 multi-turn types.

| Type | Question Example & Logical Form Pattern |
|---|---|
| single-relation | when was James Cameron born<br>$\lambda x.p(e,x)$ |
| multi-hop | what company produced film with director James Cameron<br>$\lambda x.\exists y_0 \cdots y_n.p_0\,(e,y_0) \wedge \cdots \wedge p_i\,(y_i,y_{i+1}) \wedge \cdots \wedge p_n\,(y_n,x)$ |
| multi-constraint | which movie directed by James Cameron and starred by Zoe Saldana<br>$\lambda x.p_1\,(e_1,x) \wedge \cdots \wedge p_n\,(e_n,x)$ |
| CVT | Redskins had how many losses in 1997 NFL season<br>$\lambda x.\exists y.p_0\,(e_0,y) \wedge p_1\,(y,e_1) \wedge \cdots \wedge p_n\,(y,e_n) \wedge p_{n+1}(y,x)$ |
| Yes/No | is the Kalindula a kind of guitar instrument<br>$p\,(e_1,e_2)$ |
| multi-choice | was it Bill Gates or Steve Jobs that created Microsoft<br>$\lambda x.p(e,x) \wedge ((x == e_1) \vee \cdots \vee (x == e_n))$ |
| superlative | largest lake in the world<br>$f_{\max}\,(\lambda x.p_1\,(x,e_1) \wedge \cdots \wedge p_n\,(x,e_n)\,,\lambda x.\lambda y.p(x,y),v)$ |
| comparative | rocket engine with height taller than 2.6<br>$f_{comp}\,(\lambda x.p_1\,(x,e_1) \wedge \cdots \wedge p_n\,(x,e_n)\,,\lambda x.\lambda y.p(x,y),v)$ |
| aggregation | how many movie has James Cameron directed<br>$f_{sum}(\lambda x.p(e,x))$ |
| multi-turn-entity | when was James Cameron born ### which movie he directed<br>$\lambda x_1.p_1(e,x_1)$ ### $\lambda x_2.p_2(e,x_2)$ |
| multi-turn-predicate | when was James Cameron born ### how about Bill Gates<br>$\lambda x_1.p(e_1,x_1)$ ### $\lambda x_2.p(e_2,x_2)$ |
| multi-turn-answer | when was James Cameron born ### which movie released on that day<br>$\lambda x_1.p_1(e,x_1)$ ### $\lambda x_2.p_2(x_1,x_2) \wedge \lambda x_1.p_1(e,x_1)$ |

Figure 1: Examples for each question type, including natural language question and its logical form pattern.

Next, we will introduce how single-turn questions and multi-turn questions are annotated respectively.

**Single-turn data construction**

Given a specific question type and a corresponding logical form template, we generate the <question, logical form> pairs through 4 steps including

(1) KB subgraph sampling,
(2) seed question annotation/generation,
(3) question paraphrasing/composition,
(4) logical form generation.

Figure 2 use an example to show how we annotate multi-hop questions. Other types of questions are annotated in a similar way.

The logical form of a multi-hop question $q$ has the following format:

$$\lambda x.\exists y_0 \ldots y_n\,.p_0(e,y_0) \wedge \ldots \wedge p_i(y_i,y_{i+1}) \wedge \ldots \wedge p_n(y_n,x)$$

where $e$ is an entity mentioned by $q$, $p_0,\ldots,p_n$ are predicates expressed by $q$, $y_0,\ldots,y_n$ are hidden variables, and $x$ is the answer variable.

For example, *when is the birthday of Google's founder* is a multi-hop question, whose logical form is $\lambda x.\exists y.organization\_founder(Google,y) \wedge person\_birthday(y,x)$.

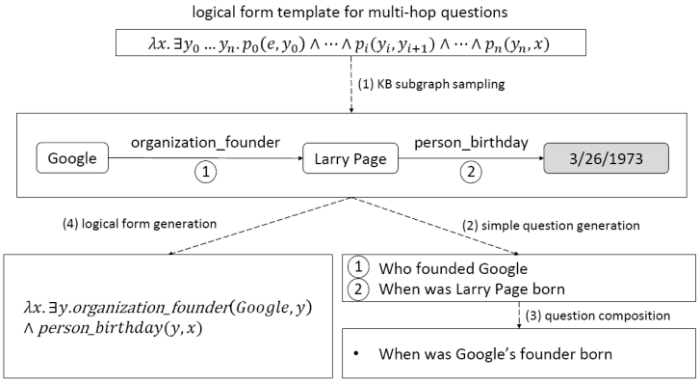


Figure 2: Multi-Hop question build workflow.

In the KB subgraph sampling step, we select valid multi-hop subgraphs from KB automatically. Each subgraph consists of two KB triples, where the object entity of the first triple is the subject entity of the second triple. Then, a template-based question generation component is used to generate two natural language questions for these two triples in the subgraph, where the answer of each generated question is the corresponding object entity. Note we annotate the seed questions for some other types by human expert instead of the QG component. In the third step, crowd sourcing annotators compose these two questions to form a multi-hop question. The annotators are also asked to paraphrase the questions to increase the diversity. Finally we translate the sampled KB subgraph into a semantic equivalent logical form, i.e., untyped lambda-calculus, and combine it with the annotated questions to compose the final <question, logical form> pairs.

**Multi-turn data construction**

In multi-turn semantic parsing task, translating a question $q_i$ into logical form $l_i$ relies on the former question $q_{i-1}$ and logical form $l_{i-1}$. Specifically, $l_i = \{x_1, x_2, \ldots, x_{|l|}\}$ contains a sub-sequence $x_{ij}$ where $x_{ij} \in l_{i-1}$ and $x_{ij}$ can not be obtained from $q_i$ directly.

In MSParS, we construct 3 types of multi-turn data according to the role of the ``lending" item $x_{ij}$, i.e., the entity, predicate and answer. Figure 3 shows the build workflow of multi-turn-entity data. In the KB subgraph sampling step, we select two triples $t_1$, $t_2$ shared a same entity ``Titanic". The two triples support two turn of semantic parsing separately. In the question annotation step an expert annotate two seed questions according to $t_1$ and $t_2$. Note that the shared entity ``Titanic" can not be used when generating the second turn question. After that, the crowd sourcing workers are asked to rewrite the seed questions, i.e., question paraphrasing. Finally, the two triples are translated into two logical forms. We combine them with the annotated questions to compose the final <$q_1$&$q_2$, $l_1$&$l_2$> pairs.

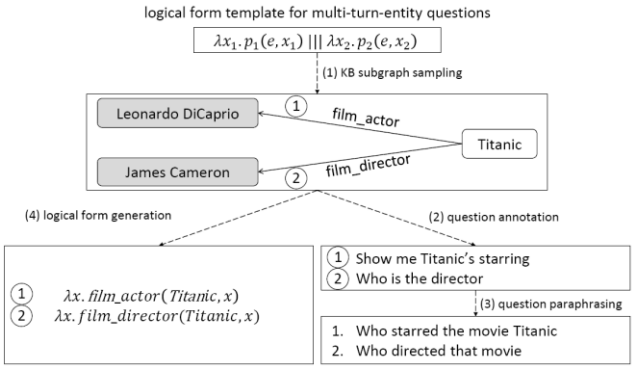


Figure 3: Multi-turn-entity question build workflow.

**Data statistic**

Figure 4 gives the statistics of MSParS, from which several characteristics can be seen. First, 41.9% questions in MSParS are single-relation questions. This is by-design, as most KB-answerable complex questions can be decomposed into a set of single-relation questions. By providing enough training data to single-relation questions, both simple questions and complex questions benefit. Second, all 81,826 questions in MSParS come from 48,917 unique question patterns based on 9,150 unique logical form patterns. It indicates that the diversity of this dataset is good, and the trained semantic parser could be robust to the paraphrases of identical meaning representations. Third, the entities occurring in MSParS are very rich as well. It makes MSParS a very challenging semantic parsing dataset. It also can be considered a new dataset for entity mention detection.

| Question Type | Statistics | | | | | | Train/Dev/Test Distribution | | |
|---|---|---|---|---|---|---|---|---|---|
| | # Q | # E | # LFP | # QP | AvgLen Q | AvgLen LF | # train | # Dev | # Test |
| single-relation | 34,316 | 18,038 | 1,256 | 25,741 | 7.4 | 9 | 26,955 | 3,727 | 3,634 |
| multi-hop | 7,452 | 1,936 | 690 | 2,043 | 10.6 | 19 | 5,938 | 780 | 734 |
| multi-constraint | 2,601 | 2,960 | 415 | 833 | 12.9 | 17 | 2,029 | 293 | 279 |
| cvt | 5,115 | 4,027 | 724 | 1,710 | 11.4 | 24 | 3,849 | 619 | 647 |
| yesno | 2,688 | 2,386 | 564 | 1,257 | 12 | 5 | 2,086 | 300 | 302 |
| multi-choice | 1,344 | 2,376 | 876 | 1,317 | 17 | 25 | 1,071 | 134 | 139 |
| aggregation | 7,710 | 6,601 | 256 | 1,649 | 9 | 10 | 5,871 | 906 | 933 |
| superlative | 8,429 | 6,506 | 222 | 2,625 | 6.3 | 26 | 6,623 | 898 | 908 |
| comparative | 357 | 254 | 48 | 168 | 8.2 | 24 | 268 | 46 | 43 |
| multi-turn-entity | 9,617 | 6,317 | 3,790 | 9,405 | 14.6 | 19 | 7,362 | 1,091 | 1,164 |
| multi-turn-predicate | 893 | 1,734 | 169 | 873 | 13.9 | 18 | 706 | 100 | 87 |
| multi-turn-answer | 1,304 | 1,287 | 140 | 1,296 | 13.3 | 29 | 1,068 | 106 | 130 |
| Overall | 81,826 | 46,733 | 9,150 | 48,917 | 9.5 | 14.7 | 63,826 | 9,000 | 9,000 |

Figure 4: Statistics and train/dev/test distribution of MSParS.

We split MSParS into three parts: train set, dev set, and test set. Generally, data in this three sets are unbiased. The distributions of question types are listed in Figure 4. The ratios of each question type are almost the same in three different datasets. Similarly, we keep the predicate frequencies of each dataset are balanced. In other words, we try to make sure that each predicate or logical form pattern is occurred in all the three datasets. The unbiased distributions benefits to verify the performance utilizing dev set when training the models on the train set. This data splitting is based on the following two rules: (1) questions sharing the same question patterns will NOT spread over different datasets. This is important as we do not want the semantic parsing to output correct results just by remembering the question patterns; and (2) if a logical form has less than three question patterns, then we will put them into train and dev sets only, instead of a test set. By doing so, every logical form pattern in the test set must occur in both train and dev sets, which makes semantic parsing evaluation reasonable.

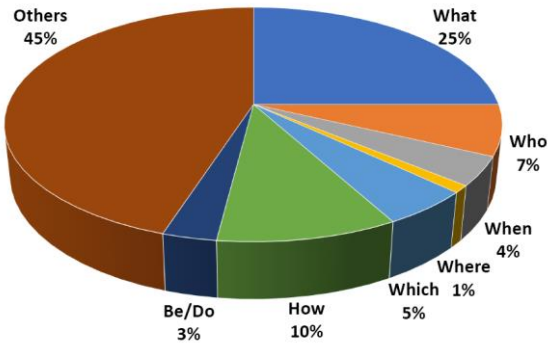


Figure 5: Question distribution of MSParS.

Figure 5 gives the question distribution based on interrogative words. Be/Do denotes the questions starting from *is*, *are*, *was*, *were*, *do*, *does*, and *did*. Others denotes questions starting from none of these interrogative words listed in the figure, which are usually keyword queries. We randomly sample 100 examples to evaluate the quality of MSParS. Quality evaluation results show that all the questions are

answerable since the logical forms are all executable, while 3% of the labeled questions contain some typo errors or spelling mistakes.

## 3 Evaluation Result

There are totally 25 teams registered for the open domain semantic parsing task, and 6 teams submitted final results. As we provide enough annotations in training data, most participating teams can fully leverage such information and achieved good results on the full test set. Therefore, we also select a hard subset for the full test set to further check the performances of different semantic parsers. Table 1 lists the rankings and scores of these 6 teams:

Table 1: Final Submissions.

| Team ID | Organization | ACC (full set) | ACC (hard subset) |
|---|---|---|---|
| Soochow_SP | 苏州大学 | 0.8568 | 0.5743 |
| NP-Parser | School of Electronics Engineering and Computer Science, PKU | 0.8373 | 0.5193 |
| WLIS | PIE Group，北京大学计算机科学技术研究所 | 0.8253 | 0.4783 |
| Binbin Deng | Fudan University | 0.6882 | 0.3541 |
| kg_nlpca_ai_lr | AI lab, Lenovo Research Institute | 0.3079 | 0.1489 |
| TriJ | 大连理工大学计算机科学与技术学院，信息检索实验室 | 0.2677 | 0.1449 |

We also check the technique reports of the first two systems. The Soochow_SP team achieves the 1st place in the open domain semantic parsing task. A transformer-based encoder-decoder framework is used for LF generation, where the number of layers in both encoder and decoder is 6. To alleviate the in-balanced question type issue, the authors proposed a synthetic training method, where new questions are generated by either replacing an entity of an original question with a new one with the same type, or replacing the type of a given entity with another valid one. The NP-Parser team uses a sketch-based method. First, LF template is selected based on each input question. Then, missing entities are filled into the LF template to form a complete an LF. Last, a seq-to-seq model is used to re-rank different LF candidates. This paper achieves the best result on the hard subset of the test set, although after the shared task deadline.

## 4 Conclusion

This paper briefly introduces the overview of this year's Open Domain Semantic Parsing shared task. We see promising results and different techniques used. We are looking forward more organizations can take part in this yearly activity, and more benchmark data sets and techniques will be delivered to the community.

## Reference


1. Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990.

2. Lappoon R. Tang and Raymond J. Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In Machine Learning: EMCL 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings, pages 466–477.
3. 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 and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, USA, August 4-8, 1996, Volume 2., pages 1050–1055.
4. Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers, pages 423–433.
5. Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1533–1544.
6. Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. CoRR, abs/1506.02075.
7. Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In The Semantic Web - ISWC 2017- 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II, pages 210–218.
8. Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 641–651.
9. Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers.
10. Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. CoRR, abs/1709.00103.