*Proceedings of the WILDRE-6 Workshop @LREC2020*, pages 68–73
Marseille, 20 June 2022


# Makadi: A Large-Scale Human-Labeled Dataset for Hindi Semantic Parsing

**Shashwat Vaibhav, Nisheeth Srivastava**
Department of Computer Science, IIT Kanpur
{shashwatv, nsrivast}@cse.iitk.ac.in

**Abstract**

Parsing natural language queries into formal database calls is a very well-studied problem. Because of the rich diversity of semantic markers across the world's languages, progress in solving this problem is irreducibly language-dependent. This has created an asymmetry in progress in NLIDB solutions, with most state-of-the-art efforts focused on the resource-rich *English* language, with limited progress seen for low resource languages. In this short paper, we present *Makadi*, a large-scale, complex, cross-lingual, cross-domain semantic parsing and text-to-SQL dataset for semantic parsing in the *Hindi* language. Produced by translating the recently introduced English language *Spider* NLIDB dataset, it consists of 9693 questions and SQL queries on 166 databases with multiple tables which cover numerous domains. It is the first large-scale dataset in the Hindi language for semantic parsing and related language understanding tasks. Our dataset is publicly available at the github repository: https://github.com/neg-loss/Makadi.git.

**Keywords:** Semantic parsing, Spider dataset, Natural Language Interface to Databases

## 1. Introduction

Semantic parsing involves mapping natural language sentences to a formal meaning representation, mainly to query an information source. It requires understanding the meaning of natural language sentences and mapping to meaning representations such as logical forms or directly into some programming language like SQL, Python, etc.

Researchers have developed several machine learning models that perform semantic parsing well in test evaluations. Recent advances in semantic parsing have improved the accuracy of neural parsers(Jia and Liang, 2016)(exact match accuracy on ATIS(Price, 1990; Dahl et al., 1994), 83.3), (Dong and Lapata, 2016)(correct answer accuracy percentage on ATIS, 84.6) and (Wang et al., 2020)(exact match accuracy on Spider(Yu et al., 2018), 65.6). These models are fuelled by the training data available to them. Other approaches to NLIDB which do not require large training data based on keywords such as NLP-Reduce(Kaufmann et al., 2007)(69.6% and 67.7% average recall and precision respectively on JOBS(Tang and Mooney, 2001)), and based on parsing such as Athena(Saha et al., 2016)(100% precision and 88.3% recall on MAS[1]) also exist. Since *English* is a resource-rich language, there exist numerous datasets for training such models, for example, Spider(Yu et al., 2018) and WikiSQL(Zhong et al., 2017). We compare our dataset with these two English language NLIDB datasets in Table 1. However, for resource-scarce languages, it becomes very difficult to find such resources, let alone have some model to tackle the problem. In this paper, we present a human-labeled dataset for cross-lingual and cross-domain semantic parsing equivalent to the existing Spider(Yu et al., 2018) dataset in Hindi mixed with the English language. We preferred working with the mixed language because it represents the real-world scenario closely as Hindi speakers generally substituting English terms into their Hindi speech rather than trying to define Hindi equivalents. We combined *Hindi* and *English* words using the *Roman script*, transliterating from Hindi to English using existing libraries (Kunchukuttan, 2020).

Creating such a dataset was challenging for many reasons. First, it wasn't always possible to find *neat Hindi equivalents* of many words from the *English* language or if they were there, they were too complicated to be used since they made look sentences not very appealing. For example, the query *"Return the average price of products that have each category code."* was translated to *"pratyek category code vaale product ka ausat price lautaen."* not translating *category* to *shrenee*. Second, table attributes present in the database were to be renamed mostly to suit the references made in the queries. For example, in most database tables, there was a column named *Name* and we translated it to *naam* such that query like *"Chocolate"* ***naam*** *ke product ka description kya hai?* stay consistent and refer to the existent database attributes. The third and the most ambiguous part was a balance of the languages used in the queries so that it represents real-world scenarios as closely as possible. For example, the query *"Find the phone number of all the customers and staff."* was translated into *"sabhee customer aur staff ke phone number ka pata lagaen."* instead of translating into *"sabhee graahakon aur karmachaariyon ke phone number ka pata lagaen."*. Another possibility related to databases was to even update values available in the database tables. For most of the cases, we chose not to update them as they are just constants and can be referred to in the same way as they were referred to in the queries in the original dataset. To the best of

[1]Microsoft Academic Search Database

our knowledge, there does not exist any such dataset for *Hindi-English mixed* language.

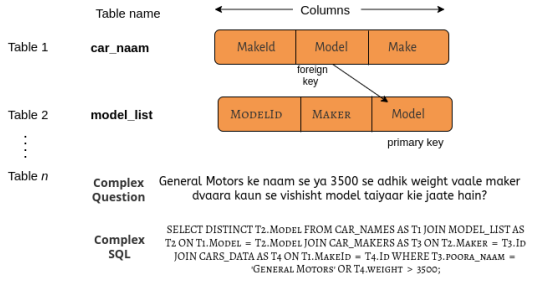


Figure 1: Our corpus contains queries of varying complexity. The above figure shows one such *Natural language query* and the corresponding *SQL* query. A Foreign key constraint is also shown in the above example.

## 2. Related Work and Existing Datasets

Several semantic parsing datasets with different queries have been created. The natural language used in these datasets is generally *English*. The meaning representations in these datasets can be in any format e.g., logical forms. Historic, monolingual datasets include GeoQuery(Zelle and Mooney, 1996), JOBS(Tang and Mooney, 2001), ATIS(Price, 1990; Dahl et al., 1994). They have been used extensively by (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Dong and Lapata, 2016; Liang et al., 2011). Other existing text-to-SQL datasets are Restaurants(Popescu et al., 2003; Tang and Mooney, 2001), Scholar(Iyer et al., 2017), Academic(Li and Jagadish, 2014), Yelp, and IMDB(Yaghmazadeh et al., 2017), Advising (Finegan-Dollak et al., 2018) containing 378, 817, 196, 128 and 131 natural language questions respectively. These datasets have been studied for decades in the NLP community(Warren and Pereira, 1982; Popescu et al., 2003; Li et al., 2006; Giordani and Moschitti, 2012; Wang et al., 2017; Xu et al., 2018; Yaghmazadeh et al., 2017).

The very large semantic parsing dataset WikiSQL(Zhong et al., 2017) contained 80,654 natural language queries and 26,521 databases in it. However, each database contained just one table and hence all the queries in it are simple. Spider(Yu et al., 2018) contains 10,181 queries in the English language with over 200 databases covering 138 domains. It contains multiple tables per database with referential constraints allowing to have complex queries in it. The recently published dataset MTOP(Li et al., 2021) consists of 100k annotated utterances in 6 languages(*English, Hindi, Spanish, Thai, French, and German*) across 11 domains for multilingual semantic parsing.

## 3. Corpus Construction

The original Spider dataset consisting of 200 databases was constructed using mainly three sources. The authors collected 40 databases from DatabaseAnswers[2] having thousands of data models across a wide range of domains. These were populated with dummy data using an online tool[3] and then manually corrected by the authors. Other 70 complex databases from SQL tutorial websites, textbook examples, and database courses were taken, and the remaining 90 databases were created from WikiSQL containing about 500 tables from different domains. The creation of this massive dataset nearly took overwhelming 1100 hours of manual effort. The authors published the dataset which included 1,034 queries in the dev set, 8,659 queries in the train set, and 488 queries in the test set. The authors chose not to make the test set public and is used to evaluate different proposed models and maintain a leaderboard[4]. Along with the queries, it even contained corresponding SQLite databases and SQL scripts to generate those databases. Also, a *table.json* file included with the dataset describes the database table attributes and the foreign key relationship existing among them. We also found that the dataset contained 7 databases with empty tables[5]. It is crucial for models like RAT-SQL(Wang et al., 2020) to have values in the tables as they use value linking. We even found out that one of the databases was missing a table from the SQL script and we took care of it.

**Our Approach** To start with, we took each statement in the *English language* from the dataset and translated it into an equivalent *Hindi language* statement. The translation process was somewhat controlled in the sense that we did not translate the queries completely(see table 2). We chose to skip translating some very commonly used *English* words and maintained a balanced extent of translation. The motivation behind such balanced translation came from the fact that Hindi speakers in general do not purely use a single language in day-to-day life. Instead, they very frequently use common words from English and sometimes even from local vernaculars. To speed up the translation process, we used *Google translate*[6]. In order to have the references made in the queries relevant, we even updated the table attributes like column names and table names suitably keeping the changes minimal. Although we had the option of translating values from the tables but we strongly avoided that because they are constants and can be referred to in a fashion similar to the original dataset. However, some of the values were changed to make the queries look more natural. For example, in the query *"sabhee mahila sankaay sadasyon ke lie pahala naam, antim naam*

[2]http://www.databaseanswers.org/

[3]http://filldb.info/

[4]https://yale-lily.github.io/spider

[5]Academic, Music, Scholar, Yelp, Geo, Restaurants, IMDB

[6]https://translate.google.com/

| Dataset | # Q | # SQL | # DB | # Domain | # Table/DB | ORDER BY | GROUP BY | HAVING |
|---|---|---|---|---|---|---|---|---|
| WikiSQL | 80,654 | 77,840 | 26,521 | - | 1 | 0 | 0 | 0 |
| **Spider** | 10,181 | 5,963 | 200 | 138 | 5.1 | 1335 | 1491 | 388 |
| **Makadi** | 9,693 | 5,294 | 166 | 133 | 5.27 | 1150 | 1261 | 313 |

Table 1: Statistics about different text-to-SQL datasets. **Makadi** is the *only* text-to-SQL dataset in code-mixed *English* and *romanized Hindi* language.

| Original Query | Google Translation | Balanced Translation |
|---|---|---|
| What are the names of the pilots in alphabetical order? | paayalaton ke naam varnaanukram mein kya hain? | pilots ke naam varnaanukram mein kya hain? |
| Which accelerator name contains substring "Opera"?? | kis tvarak naam mein "opera", sabastring shaamil hai? | kis accelerator naam mein "Opera" substring shaamil hai? |
| What is the model of the car with the smallest amount of horsepower? | sabase kam ashvashakti vaalee kaar ka modal kya hai? | sabase kam horsepower vaalee car ka model kya hai? |

Table 2: This is how we did the balanced annotation. We avoided translating queries completely.

*aur phone number dikhaen"*, the word *mahila* is value in one of the tables of *Activity* database. Keeping the consistency with the updated queries, we updated corresponding SQL queries. Since our dataset is derived from the Spider dataset, it also inherits all the qualities automatically. For example, our dataset covers a vast range of SQL patterns with the SQL components `SELECT` with multiple columns and aggregations, `WHERE, GROUP BY, HAVING, LIMIT, JOIN, INTERSECT, EXCEPT, UNION, AND, EXISTS, OR, NOT IN, LIKE` along with several nested queries. However, we faced another set of challenges while annotating the original dataset. We considered the following points.

**A) Language balance.** While creating the dataset, we felt that making a complete translation won't be the best thing to do. Because, in general, *Hindi* speakers often use *English* and this has become so normal that everybody does that. So to make our dataset more general and near-real world, we decided to keep a few English words. But now the question was about to what extent the translations were to be made. The commonality of a word in a language differs from person to person, as one person might be using some particular word than some other person. So, it was difficult on deciding which word to consider common and which word to not. To assure that the language style is consistent enough, two *Hindi speakers* did the validation and suggested improvements.

**B) Query reference.** While doing the translations, it was an absolute possibility to produce a translation in such a way that the query does not refer to the tables and columns in it at all and at the same time keep the query intent. We focused on avoiding such kinds of translations as it is natural to have references to the database in the query presented. RAT-SQL (Wang et al., 2020; Guo et al., 2019) have used the idea of *schema linking* and *value linking* extensively and have shown improvement. Another work (Bogin et al., 2019) too emphasize schema encoding and schema linking to produce the results.

**C) Missing SQL script.** We found out that in the original dataset, 7 SQL scripts were missing from the dataset. We created them ourselves. For this, we used `sqlite_web`[7] to read the corresponding SQLite databases and write the script. We also found out that several databases just had empty tables in the dataset.

## 4. Dataset Statistics and Comparison

We have summarized the statistics about our corpus already in table1. To the best of our knowledge, no such dataset is known to exist. In our dataset, we provide a dev set and a train set consisting of respectively 1,034 and 8,659 queries along with their SQL equivalents. We also ship 166 SQLite databases along with SQL scripts to generate them. We could not provide the test set and corresponding SQL queries as we did not have access to them as of writing. Since there does not exist any dataset like ours, providing a detailed comparison is a difficult task. Our dataset poses a little varied challenge to the models from what other text-to-SQL datasets present. It tests the generalization ability of the models to new and varied domains under cross-language settings.

## 5. Task Definition

With our dataset, we define a text-to-SQL task that differs from earlier text-to-SQL or more generally semantic parsing datasets which are monolingual in nature. Our dataset contains *English* transliterated natural language queries in the *Hindi* language and SQL queries distributed over vast domains. Thus, we evaluated a state-of-the-art text-to-SQL model using Hindi word embeddings on our dataset. Since our dataset contains

[7] https://github.com/coleifer/sqlite-web

| | Query Type | | | | |
|---|---|---|---|---|---|
| Component | Easy | Medium | Hard | Extra Hard | Overall |
| SELECT | 62.22(90.32) | 44.59(80.44) | 64.36(93.10) | 40.36(80.12) | 51.47(84.89) |
| WHERE | 33.63(84.01) | 30.60(72.58) | 18.18(64.89) | 15.02(46.15) | 26.09(68.67) |
| GROUP BY | 48.64(76.92) | 44.79(71.96) | 36.36(88.31) | 38.09(71.89) | 41.87(74.67) |
| ORDER BY | 49.99(73.46) | 28.77(74.66) | 55.55(83.33) | 73.07(77.70) | 52.77(77.58) |
| KEYWORDS | 78.80(89.90) | 73.49(91.79) | 64.07(84.39) | 61.16(75.30) | 70.20(86.84) |
| Query Count | 248 | 446 | 174 | 166 | 1034 |

Table 3: F1 scores for different levels of the SQL queries on our dataset. Values shown in brackets correspond to F1 scores of *GloVe* version of RAT-SQL(Wang et al., 2020) on the dev set of the Spider dataset.

mixed code i.e. *English* and *romanized Hindi*, we expected reduced performance on this dataset than English datasets. The size of the performance gap is expected to be informative about the opportunity cost of attempting to tailor semantic parsing tasks to multilingual datasets such as ours. As is conventional in such analyses, we do not include queries that require any form of common sense or knowledge from the outside world.

## 6. Evaluation Metrics

To measure the performance of different models, we consider Component matching and Exact matching as described in Spider Dataset.

**Component Matching** In a SQL query, there is a possibility of having several components for which order doesn't matter. To avoid ordering of components, we can first parse the gold SQL and decoded SQL into several sub-components and see if the same components exist in both sets.

**Exact Matching** We use the concept of component matching described above to find if the gold SQL and predicted SQL are exactly the same or not. This way equivalent gold SQL and predicted SQL will be treated the same even if some components in predicted SQL are in a different order from those in gold SQL.

**Hardness Criteria** As in the original dataset, our dataset too contained SQL queries of varying hardness. The hardness criteria depend upon a number of SQL components included in it. For example, queries consisting of just one `SELECT` component would be considered easier than the one with multiple `SELECT` components. There are four categories related to hardness: easy, medium, hard and extra hard.

## 7. Result and Analysis

To test the usability of our corpus, we evaluated our corpus on the model given by (Wang et al., 2020). We made minor modifications to it. For example, we used *Hindi* word embeddings from *FastText*[8] which also consisted of a few *English* words. We did the transliteration of the Hindi words and then used them. Since the model was not designed for a dataset like ours, unsurprisingly, it gave Exact match accuracy of 35.48% and 9.03% on easy and extra hard level queries respectively when trained for 40,000 steps. On medium and hard queries, it gave 21.97% and 14.36% exact match accuracy respectively, and overall it was found to be 21.85%. We present F1 scores of component matching in Table 3. The results described above are based on how the model performed on the dev set and not on the test set as we did not have access to the test set.

There is a clear performance gap in model performance going from English to Hindi. RAT-SQL manages to identify individual SQL clauses in natural language Hindi sentences in Makadi, leading to component matching performance of between 0.26-0.70, compared with 0.68-0.86 for similar tasks in English. However, exact match accuracy shows a much larger performance deficit, with overall accuracy at 21.85% in contrast with 59.67% ($\pm$ 2% depending on a random seed) seen in English semantic parsing.

We consider some possible explanations for this gap below. First, RAT-SQL(Wang et al., 2020) uses GloVe[9] and BERT pre-trained word embeddings for English semantic parsing, which offer comprehensive coverage for all English language words. Since a large number of English words were missing from the FastText Hindi embeddings used in our test, it is obvious to have poor performance for any model. Thus, an obvious direction of future work is to produce word embeddings for *English* and *(romanized) Hindi* words in a single vector space. Second, sometimes there exist several different transliterations for a word and in that case, which produces ambiguity in naive approaches like the one we have currently used. More sophisticated approaches could use approximate string matching techniques to map these possibilities to a single word embedding for that word. Thus, finding good semantic parsing models for mixed language datasets like Makadi offers several clear directions for improvement.

[8] https://fasttext.cc/

[9] https://nlp.stanford.edu/projects/glove/

## 8. Conclusion

In this paper, we present **Makadi** covering a variety of domains, containing complex queries in code-mixed *English* and *Hindi* language for semantic parsing and Text-to-SQL task. It is unique in the sense that it is the first such large corpus introduced in mixed language and it also happens to be the first such resource in the Hindi language. We expect that it would prove beneficial to the researchers in NLP and DB community.

## 9. Bibliographical References


Bogin, B., Berant, J., and Gardner, M. (2019). Representing schema structure with graph neural networks for text-to-SQL parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4560–4565, Florence, Italy, July. Association for Computational Linguistics.

Dahl, D. A., Bates, M., Brown, M., Fisher, W., Hunicke-Smith, K., Pallett, D., Pao, C., Rudnicky, A., and Shriberg, E. (1994). Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Dong, L. and Lapata, M. (2016). Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, August. Association for Computational Linguistics.

Finegan-Dollak, C., Kummerfeld, J. K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., and Radev, D. (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. Association for Computational Linguistics.

Giordani, A. and Moschitti, A. (2012). Translating questions to SQL queries with generative parsers discriminatively reranked. In *Proceedings of COLING 2012: Posters*, pages 401–410, Mumbai, India, December. The COLING 2012 Organizing Committee.

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.-G., Liu, T., and Zhang, D. (2019). Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535, Florence, Italy, July. Association for Computational Linguistics.

Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J., and Zettlemoyer, L. (2017). Learning a neural semantic parser from user feedback. *CoRR*, abs/1704.08760.

Jia, R. and Liang, P. (2016). Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany, August. Association for Computational Linguistics.

Kaufmann, E., Bernstein, A., and Fischer, L. (2007). Nlp-reduce: A "naıve" but domain-independent natural language interface for querying ontologies. *4th European Semantic Web Conference (ESWC 2007)*, 01.

Kunchukuttan, A. (2020). The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

Li, F. and Jagadish, H. V. (2014). Constructing an interactive natural language interface for relational databases. *Proc. VLDB Endow.*, 8(1):73–84, sep.

Li, Y., Yang, H., and Jagadish, H. V. (2006). Constructing a generic natural language interface for an xml database. In Yannis Ioannidis, et al., editors, *Advances in Database Technology - EDBT 2006*, pages 737–754, Berlin, Heidelberg. Springer Berlin Heidelberg.

Li, H., Arora, A., Chen, S., Gupta, A., Gupta, S., and Mehdad, Y. (2021). Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *ArXiv*, abs/2008.09335.

Liang, P., Jordan, M. I., and Klein, D. (2011). Learning dependency-based compositional semantics.

Popescu, A.-M., Etzioni, O., and Kautz, H. A. (2003). Towards a theory of natural language interfaces to databases. In *IUI*.

Price, P. J. (1990). Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Saha, D., Floratou, A., Sankaranarayanan, K., Minhas, U. F., Mittal, A., and Özcan, F. (2016). Athena: An ontology-driven system for natural language querying over relational data stores. *Proceedings of the VLDB Endowment*, 9:1209–1220, 08.

Tang, L. R. and Mooney, R. J. (2001). Using multiple clause constructors in inductive logic programming for semantic parsing.

Wang, C., Cheung, A., and Bodik, R. (2017). Synthesizing highly expressive sql queries from input-output examples. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2017, page 452–466, New York, NY, USA. Association for Computing Machinery.

Wang, B., Shin, R., Liu, X., Polozov, O., and Richardson, M. (2020). RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online, July. Association for Computational Linguistics.

Warren, D. H. and Pereira, F. C. (1982). An efficient easily adaptable system for interpreting natural lan-

guage queries. *American Journal of Computational Linguistics*, 8(3-4):110–122.

Wong, Y. W. and Mooney, R. (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague, Czech Republic, June. Association for Computational Linguistics.

Xu, X., Liu, C., and Song, D. (2018). SQLNet: Generating structured queries from natural language without reinforcement learning.

Yaghmazadeh, N., Wang, Y., Dillig, I., and Dillig, T. (2017). Sqlizer: Query synthesis from natural language. *Proc. ACM Program. Lang.*, 1(OOPSLA), oct.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *EMNLP*.

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

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*, UAI'05, page 658–666, Arlington, Virginia, USA. AUAI Press.

Zhong, V., Xiong, C., and Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.