Parallel Corpus Curation for Filipino Text-to-SQL Semantic Parsing

Christalline Joie Borjal[1], Moses Visperas[1], Aunhel John Adoptante[1], Ma. Teresita Abia[1],
Jasper Kyle Catapang[2], Elmer Peramo[1*]
[1]*Computer Software Division, Advanced Science and Technology Institute*
Department of Science and Technology, Diliman, Quezon City, Philippines
[2]*Department of English Language and Linguistics*
University of Birmingham, United Kingdom
*elmer@asti.dost.gov.ph

***Abstract*—Text-to-SQL models were developed over the years to allow non-technical users to interact with relational databases. Deep learning approaches require large amounts of labeled data, but the majority of the available datasets used today for natural language processing task are in English. These make text-to-SQL semantic parsing in Filipino a promising yet challenging endeavor. This research presents the iTanong corpus—a hand-labeled parallel semantic parsing corpus for Filipino Text-to-SQL tasks. The frequent code-switching or the practice of alternating between two or more languages or varieties of language in conversation and written text poses another challenge in semantic parsing for the Filipino language. The iTanong corpus contains 16,113 Filipino question and SQL pairs from two institutional databases sourced from students and employees and curated by the research team. The researchers employed Part-of-Speech tagging to guide the annotation process and analyze the various structure of the natural language queries. The usability of the corpus is tested with GPT-3 with 1,150 question-SQL pairs and achieved an execution and exact-match accuracy of 87.4% and 89.8%, respectively.**

***Index Terms*—relational databases, deep learning, semantic parsing, code-switching, Part-of-Speech Tagging, GPT-3**

I. Introduction

In today's world, where data is considered the new oil, many companies and organizations need to take advantage of data. Most of this data is stored in relational databases, requiring users to hand-labeled languages, such as SQL, to elicit information from these databases. Relational databases are widely used to store vast amounts of information represented in a grid-like structure to allow users easily fetch and draw insights from data [1]. Conveying information from this structure requires technical skills in SQL, the formal language used to interact with databases. This leaves many non-technical users at a disadvantage. Semantic parsing bridges this gap by allowing non-technical users to elicit information without learning the complexities of SQL.

Semantic parsing converts natural language sentences to meaningful and unambiguous representations [2]. It is used for intelligent question-answering systems like Photon [3] and power knowledge graph [4]. Due to its relevance, there have been a lot of previous works in this domain, evidenced by multiple extensive surveys [1] [5] [6] discussing the progress of semantic parsing. Early models implement intricate rule-based pipelines to translate user input to corresponding SQL components like the models NaLIR [7], ATHENA [8] and SQLizer [9]. Advances in artificial intelligence, like the establishment of large language models, pushed the development of neural semantic parsing. However, the researchers believed that the availability of larger datasets with which these language models can be fine-tuned could still further the development of more performant text-to-SQL systems.

Among the most experimented datasets for supervised neural semantic parsing are WikiSQL and Spider. WikiSQL contains hand-annotated examples across 24,241 tables and was released with Seq2SQL [10]. The Spider dataset was created for complex cross-domain text-to-SQL tasks [11]. Context-dependent datasets SParC [12] and CoSQL [13] were built based on Spider.

The language most often used in these benchmark datasets is English; however, researchers have begun creating benchmark datasets in other languages as well. Pilot datasets for Chinese [14] and Vietnamese [15] were studied to start the development of semantic parsing models for low-resource languages. As far as the researchers of this work are concerned, there has yet to be a research on or development of a parallel corpus for Filipino text-to-SQL semantic parsing. One of the effects of Philippine bilingualism is code-switching which usually occurs when speaking in informal situations. Code-switching from time to time gives speakers the latitude to convey their ideas with ease. In the study of Pascasio [16] where the participants were proficient in English and Filipino, it was observed that those who often code-switch are professionals, students, and employers. This, however, will pose a challenge in semantic parsing for industrial use because of the inconsistency of the grammar rules to consider. To help progress in this area, we developed a corpus of Filipino statements with SQL annotations. We trained the corpus using GPT-3 [17] to assess the semantic parsing task.

II. Related Work

A. Context-Independent Datasets

Benchmark datasets like WikiSQL and Spider are widely used for semantic parsing experiments. WikiSQL was developed along with Seq2SQL, a deep neural network for translating text to SQL queries that uses reinforcement learning. WikiSQL consists of 80,654 questions from 24,241 HTML tables. [10] It allows generalizing to unseen databases, but each query only examines one table. The Spider dataset was released as an improvement with 10, 181 questions on 200 databases. It incorporates schema complexity in its queries by allowing search on multiple tables [11]. They also shared a standard evaluation metric for semantic parsing models. The evaluation metrics are component matching, exact matching and execution accuracy.

Both of Spider and WikiSQL are utilized for context-independent semantic parsing wherein the succeeding questions do not depend on the search history. Therefore, the user intention must be clear and have enough mentions of the entities being asked in the question.

B. Context-Dependent Datasets

One realistic approach to clarify ambiguous questions is asking a series of related questions. Context-dependent semantic parsing systems take advantage of query logs and previous transactions to deliver a seamless interaction between the end-user and system. The SParC dataset/corpus was released to explore context-dependent semantic parsing. SParC forms thematically relevant interrogations for each entry selected from Spider. Overall, it consists of more than 12,000 questions which were tested using Seq2Seq with turn-level history encoder (CD-Seq2Seq) and SyntaxSQLNet with history input(SyntaxSQL-con) [12].

To advance context-dependent semantic parsing task, CoSQL was developed for dialogue systems. It contains more than 30,000 turns and 10,000 SQL annotations based on the Wizard of Oz collection. This corpus contains a substantial amount of questions that cannot be converted to SQL queries which are handled by system responses that describe the converted SQL query in comprehensible plaintext [13]. Their system employed the data with SQL-grounded dialogue state tracking, natural response generation, and user dialogue act prediction.

C. Pilot Datasets for Low-Resource Languages

Adaptations of the Spider dataset were explored for Chinese and Vietnamese to inspire further research on low-resource languages. An earlier work called CSpider manually translates all the questions to Chinese but retains the English schema. The translation is done for each database while strictly maintaining the original structure of the sentence. They used SyntaxSQLNet [18], which uses a syntax-tree-based decoder to generate complex SQL queries as a baseline model, and they modified the input embeddings to analyze character-level and word-level segmentation [14]. Two methods were explored, and in the first experiment, Glove [19] encodes the English words and Tencent [20] cross-lingual embeddings for the questions in Chinese. In the second method, the questions were represented upfront with cross-lingual embeddings.

For Vietnamese semantic parsing, Nguyen et al. [15] translated the question and schema of the Spider dataset and modified the baseline models EditSQL and IRNet. EditSQL enables context-aware semantic parsing through a query editing mechanism [21], and IRNet generates an intermediate representation called SemQL to infer SQL from domain knowledge. [22]. They enhanced IRNet with normalized pointwise mutual information (NPMI) [23] for schema linking. Additionally, latent syntactic features were used as part of the input embeddings in IRNet and EditSQL. They also used BERT-based models such as cross-lingual XLM-R-base [24] and PhoBERT-base, which is a Vietnamese RoBERTa-based embedding [25] to improve model performance. [15].

Experiments on CSpider highlight segmentation errors in word-based semantic parsers and showed that cross-lingual word embeddings yield superior results over monolingual embeddings. [14]. The Vietnamese dataset studied on extended baseline models showed improvement in exact-matching accuracy.

III. Curation Of Dataset

We explored context-independent semantic parsing for Filipino by creating a custom text-to-SQL corpus. The building process comprises data gathering, database design, sourcing of natural language questions, and SQL annotation.

A. Data Gathering and Database Design

The process for the creation of databases is shown in Figure 1. We started by collecting simple tabular data within the DOST - Advanced Science and Technology Institute (ASTI) premises and in the Computer Software Division(CSD). The DOST-ASTI is a government research and development institute having CSD as part of the internal structure where the researchers work.

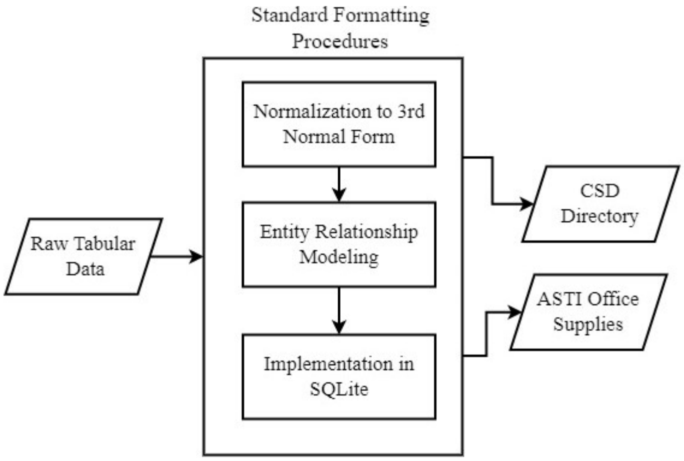

Standard Formatting Procedures
Normalization to 3rd Normal Form
Entity Relationship Modeling
Implementation in SQLite
Raw Tabular Data
CSD Directory
ASTI Office Supplies

Figure 1. Creation of iTanong Databases

A few spreadsheets of directory and inventory data office supplies were gathered. The table and column names are written inEnglish as a common practice in the industry.

The next step was normalizing the database tables to the third normal form (3NF). The research team examined the relationships between the individual attributes present in the tables to break down larger tables into smaller ones [26].

This step maintains the integrity of the database by reducing redundant information that can result in lengthy searches for a small fraction of target rows. Deletions of rows in unnormalized tables may also cause an unwanted removal of data, and updates can cause inconsistencies in information.

The first schema we called *csd* contains 18 tables and has a primary table called *Staff* wherein all connections can be traced to this table. The goal of this schema is to store employee information and assigned paraphernalia. Meanwhile, *office supplies* schema is relatively simpler, with three tables that store ASTI office supplies record.

The schemas were implemented in SQLite to simplify the connection and parsing processes. All column and table names were renamed to understandable terms and maintained language consistency from the raw source file.

B. Natural Language Questions and Structure

The respondents for our question curation task were iTanong team members and four student interns. Each one was informed about the coverage of the two schemas and was tasked to formulate questions as if inquiring within the databases. They were allowed to use English, Tagalog and Taglish, variants of the Filipino language. We have gathered 16,113 questions from both databases.

A manual inspection was performed to identify unanswerable questions. These statements are either too vague for the SQL annotators or incomplete, requiring external knowledge. Asking *Ilang percent ng employees ang fresh graduates?* (How many percent of employees are new graduates?) is tagged as unanswerable for these reasons. Intuitively, the term ”fresh graduates” refers to the employees who have been hired in the same year upon receiving their diplomas which is not an entry on the database.

Another concern is question specificity is *Lahat ba ng employee ay full-time?* (Are all employees full-time?). The word *full-time* may be interpreted in many forms, but in the context of retrieving CSD information, we take the keyword ”full-time”, and expound further as ”full-time *nagtatrabaho*”(full-time workers). In the Filipino work ethic, *full-time* refers to tenured workers. Employment status is represented in the *status id* attribute of the *Staff* table but the word full-time is not an entry in the linking table; instead, the term ”regular” is registered. We limited the scope to those questions that can be converted to SQL SELECT queries, but some questions sourced from the respondents are answerable by yes or no (Boolean value). We did not include these questions in our corpus.

Using part-of-speech (POS) tagging on each collected question reveals a generalized phrasal structure pattern. This is the process of assigning a POS tag for each token in a sentence. The part-of-speech tag is a crucial part of parsing, wherein it is used to find named entities in information extraction [27]. For this reason, it served as a guide to disambiguate a piece of text for SQL annotation, as words tend to have several meanings depending on the context.

Shown in Figure 2 is the process used for extracting the POS patterns. The Filipino questions were first translated to English as an additional step for code-switching and to compare the patterns for each language. Overall, two sets of questions underwent this process. Punctuation marks were removed before tagging. The Stanford POS Tagger was used for the English questions through Spacy [28]. The POS extracted followed the 45-tag Penn Treebank tagset [29] for English. For Filipino, SMTPOST [30] based on the MGNN 230-tag tagset was used.

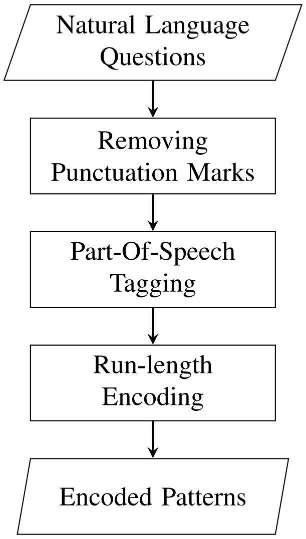


Figure 2. Extraction of POS Patterns

Run-length encoding of the pattern follows the POS tag extractions. In run-length-encoding, if an item occurs *n* times repeatedly, the *n* occurrences are replaced with *nd* [31] where *d* is the number of times the character occurred. For the phrase *”project assignment ni Christalline Joie Borjal”*, the pattern returned by SMTPOST is `FW-FW-DTP-NNP-NNP-NNP`. With run-length encoding it is simplified to `FW2-DTP-NNP3`. Only the characters were used to cluster similar patterns. The `FW` tag which refers to a foreign word was manually compared to the English POS tag.

We also translated the Tagalog and Taglish questions to English to explore code-switching for future works. Tables I and II show the top 5 extracted patterns in English and Filipino respectively. For the English patterns, it is observed ranks 1 – 3 that the curators opted to phrase their inquiries directly in the form of noun phrases. The tags `NN`, `IN` and `NNP` pertain to noun, preposition or subordinating conjunction and proper noun respectively as documented in the Penn tagset. This shows that as long as the required entities are mentioned, a table response is expected at the end of the curator. Ranks 4-5 are formed in a Wh-pronoun interrogative represented by `WP` followed by a third-person verb in present tense `(VBZ)`, and a determiner `(DT)` which modifies the noun phrase.

TABLE I
FREQUENCY COUNT OF TOP 5 ENGLISH POS PATTERNS

	English Patterns	Count
1	NN IN NN NNP	957
2	NN IN NN	673
3	NN NNP IN NN NNP	550
4	WP VBZ DT NN IN NN NNP	548
4	WP VBZ DT NN IN NN	366

The English sentence structure is phrased similarly to the Filipino source question but code-switching is evident in the Filipino dataset shown in Table II. It is indicated by the `FW` tag which means *foreign word*. The `FW` tag may indicate mentions of English, Spanish and Latin words according to the MGNN tagset. Right now, the possibility that the encoded `FW` tag may refer to a single word or a foreign phrase is not accounted. It is evident that code-switching is unavoidable on work-related inquiries based on the results. There are also POS tags for the Filipino language such as `CCB`, `CCP` and `PRQ` . These tags include `CCB` which is a conjunction variant `CCP` for ligatures like *na* and `PRQ` for singular interrogatives.

TABLE II
FREQUENCY COUNT OF TOP 5 FILIPINO POS PATTERNS

	Filipino Patterns	Count
1	NNP CCB FW NNP	1889
2	PRQ _CCP FW NNPA DTC FW NNP	1394
3	PRQ _CCP FW CCB FW NNP	1394
4	NNP NNPA CCB FW NNP	1386
4	FW CCB FW NNP	833

This research focuses on the construction of simple and complex SELECT commands. The SQL component coverage was adapted from the identified keywords in Spider such as `SELECT`, aggregations, `WHERE`, `GROUP BY`, `HAVING`, `ORDER BY`, `LIMIT`, `JOIN`, `INTERSECT`, `EXCEPT`, `UNION`, `NOT`, `IN`, `OR` , `AND`, `EXISTS`, `LIKE` and nested queries [11]. The extracted POS patterns were used to determine the columns requested, the conditions to be met, and the operations to do.

Simple queries retrieve data from one table only and follow the base structure `SELECT` [*col*] `FROM` [*tab*] `WHERE` [*cond*]. This type of question is identified if the entities mentioned in the query belong to the same relational table. As shown in Table III, entities *iTANONG* and *Project* are mentioned, which both relate to the *Project* table. The curator used *Tungkol*, which can be interpreted the same as the word ”about the” referring to the noun phrase *ang iTANONG Project*. The base SQL structure is plugged with the corresponding values.

TABLE III
ANNOTATION OF SIMPLE SQL QUERIES

sentence	Tungkol saan ang iTANONG Project?
patterns	RBR PRQ DTC NNP NNP

SELECT description FROM project
WHERE projectname=’iTANONG’;	

Table IV shows complex queries that involve joining two or more tables. The question mentions the entities *staff* and *project* which are two separate but connected relations in the database. Breaking down the sentence, *bilang ng* which is a term referring to quantity is mapped to the aggregator `COUNT` while the term *staff kada project* means that the counting of staff should be done for every project. The `GROUP BY` statement groups the result set.

TABLE IV
ANNOTATION OF COMPLEX SQL QUERIES

sentence	Ilista ang bilang ng staff kada project
patterns	VBTF DTC NNC CCB NNC PRI FW

SELECT t3.projectname, count(*) FROM staff AS t1
JOIN stafftoproject AS t2 on t1.staffid = t2.stp_staffid
JOIN project AS t3 on t2.stp _projectid = t3.projectid
GROUP BY t3.projectname ;	

The need for aggregation operators is spotted by finding the words such as *bilang*, *ilan*, and *kabuuan* followed by a noun phrase. The conditions for the `WHERE` clause is not always explicitly defined, so the annotators need to analyze whether the entities mentioned in the question is an attribute, value or table name. This research focuses only on the annotation of commonly used SQL statements across all dialects. Mapping of other SQL functions are left for further research.

IV. STRUCTURE-BASED SCHEMES FOR IDENTIFYING TRENDS IN ANSWERABLE NATURAL LANGUAGE QUESTIONS

In reality, natural language questions tend to set aside formal grammatical rules, which results in the difficulty in annotating with restricted logical forms like SQL. Since we plan to employ our corpus in developing a Filipino Text-to-SQL semantic parsing system in our future researches, we explored two schemes to analyze the natural language question structures. Our prior manual inspection guides us to assume that the curated questions follow unambiguous syntax. Using the run-length-encoded POS tags as the basis, we find variations by inspecting the sequence length and local sequence alignment among different patterns. The following schemes were tested on 16, 113 data points from question structures on two databases.

A. Sequence Length

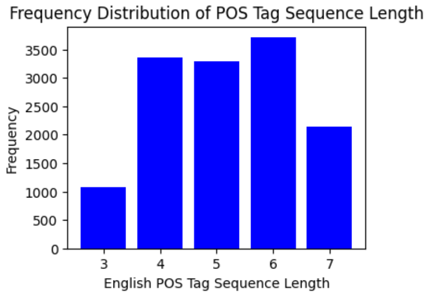


Figure 3. Statistics of English POS Sequence Length

We compute the sequence length by counting each POS tag in the run-length-encoded POS pattern. The figures 3 and 4 shows the statistics of the top five varying lengths for English and Filipino POS tags respectively. Evidently in both cases, the lengths 4,5 and 6 recorded significantly higher frequencies when compared to extremely short or longer POS patterns. Overall, this may be attributed to the intent of conveying inquiries using just the adequate amount of words possible resulting in short run-length-encoded POS patterns.

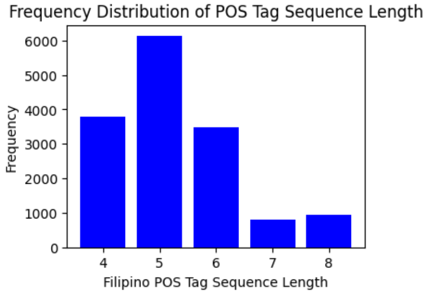
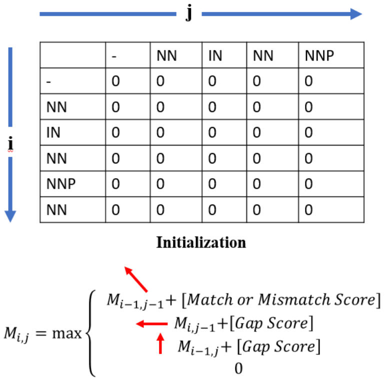


Figure 4. Statistics of Filipino POS Sequence Length

B. Local Sequence Alignment

Observing the variations of new questions may tell about the reliability of the data on hand or the quality of any incoming data points. In this regard, the Smith-Waterman algorithm was implemented to get the local sequence alignment of encoded POS patterns which serves as a monitoring scheme for similarly formed questions by comparing a given pattern to those previously seen in the corpus. Prior to local alignment, extracting the unique run-length-encoded POS patterns from the original data limits the data points for this process to 1,562 and 617 in English and Filipino respectively.

Initializing the matrix for local sequence alignment involves representing the POS patterns with zero scores as depicted in Figure 5. In this picture, the patterns `NN-IN-NN-NNP` and `NN-IN-NN-NNP-NN` are being aligned to each other. Scoring schemes for *match*, *mismatch* and *gap score* determine the direction of traversal in finding same segments of two patterns. Figure 6 shows an example of the optimally aligned pattern's traceback.

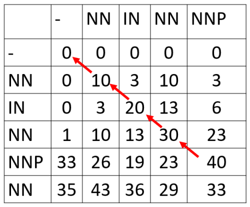


	-	NN	IN	NN	NNP
-	0	0	0	0	0
NN	0	0	0	0	0
IN	0	0	0	0	0
NN	0	0	0	0	0
NNP	0	0	0	0	0
NN	0	0	0	0	0

Initialization

$$M_{i,j} = \max\begin{cases} \nwarrow\ M_{i-1,j-1} + [Match\ or\ Mismatch\ Score] \\ \leftarrow\ M_{i,j-1} + [Gap\ Score] \\ \uparrow\ M_{i-1,j} + [Gap\ Score] \\ 0 \end{cases}$$

Figure 5. Initialization of the Smith-Waterman Algorithm



	-	NN	IN	NN	NNP
-	0	0	0	0	0
NN	0	10	3	10	3
IN	0	3	20	13	6
NN	1	10	13	30	23
NNP	33	26	19	23	40
NN	35	43	36	29	33

Figure 6. Traceback of optimal alignment

V. Experiments

The researchers investigate the usefulness of iTanong corpus by fine-tuning a GPT-3 model with the OpenAI API. However, due to the cost of API calls, only 1,150 question pairs, each from Filipino and English were selected for training. The selection process involves picking questions with SQL annotations that in totality cover most table and attributes of the database schema. The researchers believe that including the English examples makes up for the code-switching.

Generative Pre-Trained Transformer (GPT-3) [17] is the third iteration of OpenAI's powerful pre-trained generative models. It can solve a wide variety of task from classification, summarization, completion, code generation, and many more by zero-shot, multi-prompt, or fine-tuning a base model. The GPT-3 model was trained with widely scraped internet data making it proficient with English texts, but still capable of text generation for some low-resource languages like Filipino.

Fine-tuning a GPT-3 model relies heavily on the quality of prompts, which would force the researchers to concentrate on prompt engineering. The process starts by adding specialized tokens at the end of each question and SQL code to mark the end of the query. The phrase *"SQLite for:"* was inserted in front of the question to instruct the GPT3 engine on what task to do. The dataset was then converted into a `jsonl` file as instructed by the API.

```
prompt: " SQLite for: Question ->\n"
completion: " Query [EOS]"
```

The `davinci` engine –the most powerful engine available – was used as the base for the text-to-SQL model and was only trained for one epoch. For inference, the fine-tuned model used a `temperature` of 0.30, `max_tokens` of 500, `frequency_penalty` of 0.15, `presence_penalty` of 0.0 and stop token `[EOS]`.

VI. Results and Future Work

The fine-tuned model was tested using 285 Filipino natural language questions. The execution accuracy and exact match accuracy were computed using the official evaluation script provided by *Spider*. As shown in Table V, the model achieved an execution accuracy of 0.874 and exact match accuracy of 0.898. The fine-tuned model achieved a relatively high score even though the `davinci` engine was mostly trained using English queries.

TABLE V
Performance of Fine Tuned Davinci on Tagalog Dataset

	Easy	Medium	Hard	Extra	Total
Count	180	93	3	9	285
Exec	0.944	0.796	0.333	0.444	0.874
Count	180	93	3	9	285
Match	0.978	0.817	0.333	0.333	0.898

The GPT-3 architecture model is pre-trained on broad internet data which are often dominated by English texts and one factor we considered to address the code-switching in the training data. The results may also be attributed to the findings in Tables II and III on the summary of English and Filipino POS structures regarding the clarity of questions being asked by the respondents. By eliminating ambiguity in the question, mapping the relevant entities to SQL became easier both for the annotators and the semantic parsing models in predicting queries with Easy and Medium difficulty. Low performances on more complex queries were observed. Hard queries got 0.333 for both execution and exact match accuracy and Extra Hard queries attained 0.444 and 0.333 for execution and exact match accuracy respectively. The low sample size for training and testing posed an impact on the accuracy. Increasing the sample size will improve the performance of the model.

One limitation of the study is that the respondents for the curation of questions are people within the project. This entails that bias in the curation process cannot be ruled out. It is highly recommended to gather questions from end-users or people outside the project to cater to the diversity of questions that may be asked.

While the researchers shared insights in labeling code-switched inquiries, the inner mechanism of the models in handling code-switching was not covered. Future research should also investigate code-switching further in developing Text-to-SQL systems which was challenging enough to address in the SQL annotation.

VII. Conclusion

This research introduces a proof-of-concept Filipino Text-to-SQL semantic parsing dataset to further the development of low-resource languages. The creation of databases, curation of natural language questions, and SQL annotation were covered in the study. The semantics of natural language questions was mapped to SQL using Part-of-Speech tagging as a guide, which was also discussed in detail.

We hope that sharing the iTanong corpus curation process will pave a way to encourage developments for low-resource languages and advances in exploring code-switching in Text-to-SQL models.

Acknowledgment

The researchers would like to express their most profound appreciation to the unwavering support of the iTanong staff, student interns, the Computer Software Division, and the Department of Science and Technology - Advanced Science and Technology Institute (DOST-ASTI) for making this research possible.

References

[1] Bowen Qin, Binyuan Hui, Lihan Wang, Min Yang, Jinyang Li, Binhua Li, Ruiying Geng, Rongyu Cao, Jian Sun, Luo Si, Fei Huang, and Yongbin Li. A survey on text-to-sql parsing: Concepts, methods, and future directions, 2022.

[2] Raymond J. Mooney. Learning for semantic parsing. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Proceedings of the 8th International Conference (CICLing 2007)*, pages 311–324, Mexico City, Mexico, February 2007. Springer: Berlin, Germany. Invited paper.

[3] Jichuan Zeng, Xi Victoria Lin, Steven C.H. Hoi, Richard Socher, Caiming Xiong, Michael Lyu, and Irwin King. Photon: A robust cross-domain text-to-SQL system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 204–214, Online, July 2020. Association for Computational Linguistics.

[4] Yachen Tang, Haiyun Han, Xianmao Yu, Jing Zhao, Guangyi Liu, and Longfei Wei. An intelligent question answering system based on power knowledge graph. *CoRR*, abs/2106.09013, 2021.

[5] Shanza Abbas, Muhammad Umair Khan, Scott Uk-Jin Lee, Asad Abbas, and Ali Kashif Bashir. A review of nlidb with deep learning: Findings, challenges and open issues. *IEEE Access*, 10:14927–14945, 2022.

[6] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, and Hongrae Lee. Natural language to sql: Where are we today? *Proc. VLDB Endow.*, 13(10):1737–1750, mar 2021.

[7] Fei Li and Hosagrahar Jagadish. Nalir: An interactive natural language interface for querying relational databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 06 2014.

[8] Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish Mittal, Diptikalyan Saha, and Karthik Sankaranarayanan. Athena++: Natural language querying for complex nested sql queries. *Proc. VLDB Endow.*, 13(12):2747–2759, sep 2020.

[9] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: Query synthesis from natural language. *Proc. ACM Program. Lang.*, 1(OOPSLA), oct 2017.

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

[11] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *CoRR*, abs/1809.08887, 2018.

[12] Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir R. Radev. Sparc: Cross-domain semantic parsing in context. *CoRR*, abs/1906.02285, 2019.

[13] Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander R. Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter S. Lasecki, and Dragomir R. Radev. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. *CoRR*, abs/1909.05378, 2019.

[14] Qingkai Min, Yuefeng Shi, and Yue Zhang. A pilot study for chinese SQL semantic parsing. *CoRR*, abs/1909.13293, 2019.

[15] Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. A pilot study of text-to-sql semantic parsing for vietnamese. *CoRR*, abs/2010.01891, 2020.

[16] Emy M. Pascasio. The filipino bilingual from a sociolinguistic perspective. *Philippine journal of linguistics*, pages 69–79, 2003.

[17] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[18] Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir R. Radev. Syntaxsqlnet: Syntax tree networks for complex and cross-domaintext-to-sql task. *CoRR*, abs/1810.05237, 2018.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014.

[20] Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. pages 175–180, 01 2018.

[21] Rui Zhang, Tao Yu, Heyang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir R. Radev. Editing-based SQL query generation for cross-domain context-dependent questions. *CoRR*, abs/1909.00786, 2019.

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

[23] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of the Biennial GSCL Conference 2009*, 01 2009.

[24] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019.

[25] Dat Quoc Nguyen and Anh Tuan Nguyen. Phobert: Pre-trained language models for vietnamese. *CoRR*, abs/2003.00744, 2020.

[26] Toby J. Teorey, Sam S. Lightstone, Tom Nadeau, and H. V. Jagadish. *Database Modeling and Design: Logical Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2011.

[27] James H. Martin Daniel Jurafsky. *Speech and Language Processing*. 2019.

[28] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020.

[29] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[30] Nicco Nocon and Allan Borra. SMTPOST using statistical machine translation approach in Filipino part-of-speech tagging. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 391–396, Seoul, South Korea, October 2016.

[31] David Salomon. *Data Compression The Complete Reference*. Springer-Verlag New York, Inc, 3rd edition, 2004.