[User Study] Questionnaire

Hello,

Thank you for participating our case study on "MT-Teql: Evaluating and Augmenting Neural NLIDB on Real-world Linguistic and Schema Variations". Neural NLIDB is a natural language interface to database which translate human utterances into SQL queries by neural models. In general, NLIDB take human utterance and database schema as input and generate SQL queries as output. MT-Teql aims to evaluate NLIDBs by a set of Metamorphic Relations (MR). For instance, we may change the prefix of human utterance, e.g., "what is the average age of student?" -> "tell me the average age of student.". Under the prefix change, NLIDBs are expected to give consistent outputs.

In this case study, please assume yourself as an NLIDB developer who wish to adapt an NLIDB for your own application scenario based on a collection of existing general-purpose NLIDBs. First, you need to confirm the capability reported by MT-Teql is correlated with its MR. Then, you check the usefulness of each capability from your own experience. After the confirmation, you further give rank to five NLIDBs based on the result reported by MT-Teql. Finally, we will give the application domain and desired capability of your NLIDB. You are encouraged to add new MR for your own scenario and desired capability. For those who are not familiar to Metamorphic Relations, please refer to the following introduction or contact us for a short user education.

Thank you for your participation.

Best,
The author of MT-Teql

[Introduction to MR]

Metamorphic Relations can be deemed as software invariant property. For instance, for sin() function in C++, we know that for any input sin(x) = sin(x+360). Applying this property, we can automatically generate a lot of test cases without knowing the expected output of individual inputs. In the case of NLIDB, we can easily derive a MR where different prefixes (e.g., what is, tell me, let me know, ...) in human utterance should not affect the output of NLIDB. Applying this MR, we can transform utterance in the dataset and check the output of NLIDB given original utterance and transformed utterance.

[Participant demographic information]

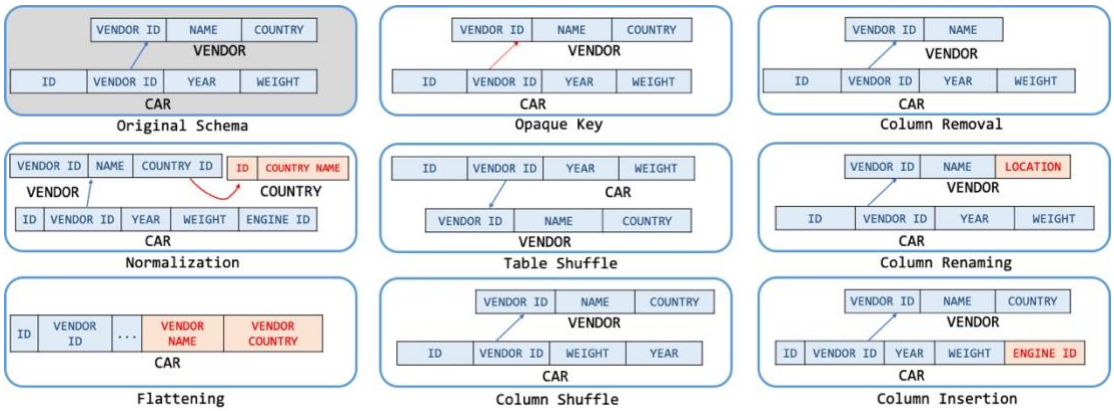
1. Did you at least take one graduate Database, NLP (including general AI) or Software Engineering course or have experience on developing DB, NLP/AI or SE-related software?
   - Yes
   - No

P1	P2	P3	P4	P5	P6	P7
Yes	Yes	Yes	Yes	Yes	Yes	Yes

[Confirmation on the correlation between MR and capability]

MR describe a way to generate test cases and capability gives human-readable description on the desired property of MR. You need to give the score on the correlation between MR and capability. Higher score denotes stronger correlation.

1: they are not correlated;
2. they share a little correlations;
3. they are correlated.



2. Do you feel the MRs and the derived Capability is correlated.

MR 1 [Prefix Insertion (PI)]
what is the age of all singers? -> [tell me] what is the age of all singers?

MR 2 [Prefix Removal (PR)]
what is the age of all singers? -> the age of all singers.

MR 3 [Prefix Substitution (PS)]
what is the age of all singers? -> tell me the age of all singers?

Capability:
Robust for users from diverse linguistic background who start utterance with diverse prefixes.

- 1
- 2
- 3

P1	P2	P3	P4	P5	P6	P7
2	2	3	3	3	2	2

3. Do you feel the MRs and the derived Capability is correlated.

MR 4 [Synonym Substitution]
what is the number of singers? -> what is the amount of singers?

Capability:

Robust for users from diverse linguistic background who indicate aggregates in diverse forms.

□ 1

□ 2

□ 3

P1	P2	P3	P4	P5	P6	P7
3	2	3	3	3	2	3

4. Do you feel the MRs and the derived Capability is correlated.

MR 5 [Column Renaming]

rename a column name by its synonym.

Please refer to "Column Renaming" for illustrative examples.

Capability:

Robust for users from diverse linguistic background who express an attribute in diverse ways.

□ 1

□ 2

□ 3

P1	P2	P3	P4	P5	P6	P7
2	2	3	3	2	3	2

5. Do you feel the MRs and the derived Capability is correlated.

MR 6 [Normalization]

normalize a column into a new reference table.

MR 7 [Flatten]

flat a reference table based on foreign key constraints.

Please refer to "Normalization" and "Flatten" for illustrative examples.

Capability:

Robust for schemas with diverse design styles which follow diverse normal forms.

□ 1

□ 2

□ 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	3	2	3	2

6. Do you feel the MRs and the derived Capability is correlated.

MR 8 [Opaque Key]
randomly remove foreign key from schema.

Please refer to "Opaque Key" for illustrative examples.

Capability:
Robust for schemas with diverse design styles which lack foreign key constraints.

- 1
- 2
- 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	2	1	2	1

7. Do you feel the MRs and the derived Capability is correlated.

MR 8 [Table Shuffle]
change the order of tables stored in schema.

MR 9 [Column Shuffle]
change the order of columns stored in table.

Please refer to "Table Shuffle" and "Column Shuffle" for illustrative examples.

Capability:
Robust for schemas with diverse design styles which stores in diverse orders.

- 1
- 2
- 3

P1	P2	P3	P4	P5	P6	P7
3	2	3	3	1	2	1

8. Do you feel the MRs and the derived Capability is correlated.

MR 10 [Column Removal]
remove a column (which is not mentioned by user) from a table in the schema.

MR 11 [Column Insertion]
add a column to a table in the schema. we use knowledge graph to ensure the inserted column is valid and natural.

Please refer to "Column Removal" and "Column Insertion" for illustrative examples.

Capability:
Robust for schemas with diverse design styles who contains/misses irrelevant columns.

- 1
- 2
- 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	3	2	3	1

[Checking the usefulness of capability]

Do you feel the capability is an important property for your NLIDB. You need to give the score on the usefulness on each capability. Higher score denotes higher usefulness.

1: it is not useful;
2. it is good to have;
3. it is very useful for me.

9. Is it import for an NLIDB to be "robust for users from diverse linguistic background who start utterance with diverse prefixes"?
   - □ 1
   - □ 2
   - □ 3

P1	P2	P3	P4	P5	P6	P7
2	3	2	3	2	3	2

10. Is it import for an NLIDB to be "robust for users from diverse linguistic background who indicate aggregates in diverse forms"?
   - □ 1
   - □ 2
   - □ 3

P1	P2	P3	P4	P5	P6	P7
2	3	2	3	3	2	2

11. Is it import for an NLIDB to be "robust for users from diverse linguistic background who express an attribute in diverse ways"?
   - □ 1
   - □ 2
   - □ 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	3	3	1	2

12. Is it import for an NLIDB to be "robust for schemas with diverse design styles which follow diverse normal forms"?
   - □ 1
   - □ 2
   - □ 3

P1	P2	P3	P4	P5	P6	P7
2	3	3	3	3	2	1

13. Is it import for an NLIDB to be "robust for schemas with diverse design styles which lack foreign key constraints."?
   - □ 1
   - □ 2

□ 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	3	2	1	1

14. Is it import for an NLIDB to be "robust for schemas with diverse design styles which stores in diverse orders"?
    □ 1
    □ 2
    □ 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	3	1	2	3

15. Is it import for an NLIDB to be "robust for schemas with diverse design styles which contains/misses irrelevant columns"?
    □ 1
    □ 2
    □ 3

P1	P2	P3	P4	P5	P6	P7
3	3	3	3	1	2	1

[Ranking based on Evaluation Result of MT-Teql]

Capability	SSN	IRN	GGL	RAT	Duo
For users from diverse linguistic background.	😑	🙂	😑	🥳	🥳
who start with diverse prefixes.	🙂	🙂	🥳	🥳	😑
who indicate aggregates in diverse forms.	🥳	🙂	😑	😑	🥳
who express an attribute in diverse ways.	🙂	🙂	🥳	🥳	😑
For schemas with diverse design styles.	🥳	😑	🙂	🥳	🙂
which follow diverse normal forms.	🥳	😑	🙂	🥳	🙂
which lack foreign key constraints.	🥳	🙂	🙂	😑	😑
which stores in diverse orders.	😑	🙂	🥳	🥳	🙂
which contains/misses irrelevant columns.	🥳	😑	🙂	🥳	🙂
Overall Robustness	😑	🙂	😑	🥳	🙂
Overall Accuracy	🙂	😑	😑	😑	🥳

The table reports the evaluation result of MT-Teql. SSN, IRN, etc., denote different NLIDBs.

Please submit the your own preference based on above table.
□ is good; 😣 is bad; and 😐 is neutral.

	P1	P2	P3	P4	P5	P6	P7
SSN	😐	□	😐	😣	😐	😐	😐
IRN	😐	😣	😣	😣	😣	😣	😣
GGN	😐	😣	😐	😐	😐	😐	😐
RAT	□	□	□	□	□	□	□
Duo	□	😣	😣	😣	😐	😐	😐

[Adapt MT-Teql to your own domain]
Please try to design MRs that can be used to evaluate the given capability under given domain. You are encouraged to give more than one MR for one question.

[Sample Question]

Assume your target users are not mature on designing a correct database schema. Therefore, they may forget to add primary key constraint in a table. You wish to evaluate whether your NLIDB is robust no matter whether the primary key exists in the schema.

[Sample MR]

We can check the output of NLIDB by randomly removing the primary key constraints from schema and see if the output is consistent.

Assume your target users are from many non-English speaking countries. Therefore, your NLIDB accepts multi-lingual utterance. You wish to evaluate whether your NLIDB is robust on diverse language.

P1: We can check the output of NLIDB by asking the same question in different languages and see if the output is consistent.
P2: We can check the output of NLIDB by adding multilingual utterance and observe if the output is right.
P3: We can check the output of NLIDB by 1) randomly replacing a word in natural language description with its synonym, or 2) randomly replacing/deleting/adding a letter.
P4: Construct the inputs with same or opposite meaning for different languages and see if the result is expected.
P5: Assume we have reliable translatior (like google translation, bing translation), we can translate the non-English setence S1 to English sentence S2 and then compare the results of NLIDB. S1 and S2 should have similar corresponding SQL statements.
P6: N/A
P7: We can check the output of NLIDB by a large number of different people using diverse language and see if the output is consistent.

Assume the database designer and target users is different. Therefore, users may not able to explicitly mention a column. For instance, users may ask about "surname" or "family name" while there is only "last name" in the database. You wish to evaluate whether your NLIDB is robust that no matter how the users mention the database column.

P1: We can check the output of NLIDB by asking through different inexplicit attributes.
P2: We can check the output of NLIDB by randomly asking different database column and see if the output is consistent.
P3: We can check the output of NLIDB by 1) randomly renaming a column in DB, or 2) randomly replacing a word in natural language description with its synonym.

P4: Generate inputs with similar or different meanings to the ground truth, and evaluate whether the output is expected.
P5: Given the column name, try to design sentences with synonyms (every words in the sentences are the same except the column name, all the sentences should have similar NLIDB outputs.
P6: N/A
P7: We can query one column by using different words, which are all refer to the same column and account the number of successful query.