

Text-to-SQL Generation For Question Answering On Electronic Medical Records

Ping Wang, Tian Shi, Chandan K. Reddy

Department of Computer Science

Virginia Tech, Arlington, VA 22203

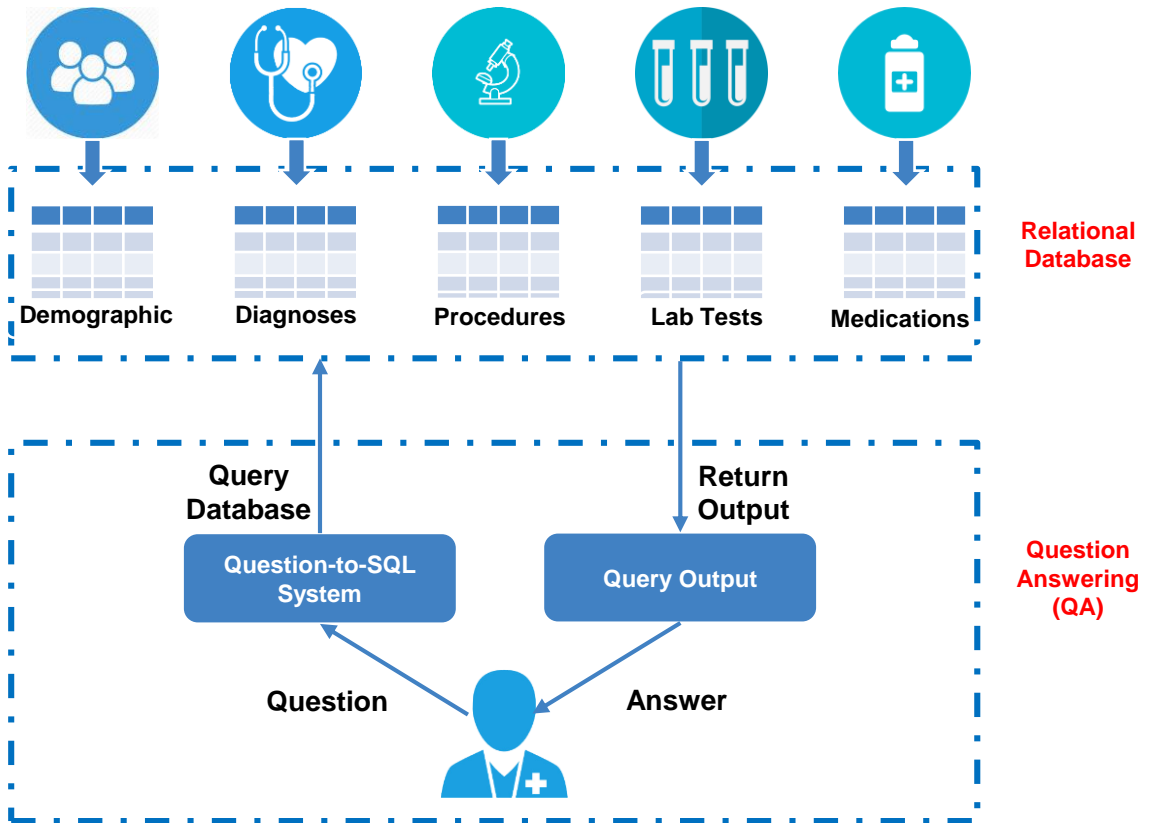


VirginiaTech

Outline

- Introduction
- Challenges
- Our Contributions
- Experiments
- Conclusion

Introduction



QA Examples on Electronic Medical Records

Relational Database

LAB TEST	SUBJECT ID	HADM ID	TEST LABEL	TYPE	DIAGNOSES	SUBJECT ID	HADM ID	SHORT TITLE
	66411	178264	Anion gap	Chemistry		9258	183354	Cardiogenic shock
	66411	178264	Hematocrit	Hematology		9258	183354	Convulsions NEC
	29961	196409	Bicarbonate	Chemistry		66411	178264	Atrial fibrillation
	29961	196409	Free calcium	Blood gas		66411	178264	Ac diastolic hrt failure
PRESCRIPTIONS	SUBJECT ID	HADM ID	DRUG	TYPE	PROCEDURES	SUBJECT_ID	HADM_ID	SHORT TITLE
	66411	178264	5% Dextrose	Base		9258	183354	Procedure-one vessel
	66411	178264	Miconazole powder 2%	Main		9258	183354	Insert endotracheal tube
	29961	196409	Warfarin	Main		66411	178264	Abdomen artery incision
	29961	196409	Iso-Osmotic	Base		66411	178264	Venous catch NEC
DEMOGRAPHIC	SUBJECT ID	HADM ID	NAME	AGE	ADMISSION TYPE	DAYS	PRIMARY DISEASE	ADMISSION LOCATION
	990	184231	Dawn Brill	89	EMERGENCY	7	Gastrointestinal bleed	Emergency room admit
	17772	122127	Ruben Thomas	0	NEWBORN	6	NEWBORN	Clinic referral/premature
	66411	178264	Jennifer Fulgham	72	EMERGENCY	8	Femoral artery thrombosis	Emergency room admit
	29961	196409	Mary Corson	72	EMERGENCY	20	S/P fall	Clinic referral/premature

Question samples

Q1: Tell me the primary disease and prescribed medicines of patient Mary Corson.

A1: S/P fall; Warfarin, Iso-Osmotic.

Q2: Give me the titles of procedures underwent by patient Jennifer Fulgham.

A2: Abdomen artery incision, Venous catch NEC.

Q3: How many patients were admitted at emergency room?

A3: 2

Challenges

- **Question generation:**
 - ◆ **No existing dataset** for question answering on relational database in healthcare domain.
 - ◆ **Not feasible to enumerate all possible questions** on the database.
- **Question-to-SQL generation:**
 - ◆ **Multiple columns**
 - ◉ What is the admission time and primary disease of Michael Tucker?
 - ◆ **Multiple tables and table unawareness**
 - ◉ Provide the primary disease of Michael Tucker and the procedure names he underwent.

Our Contributions

- **Create a large-scale MIMICSQL dataset:**
 - ◆ For Question-to-SQL generation task in **healthcare** domain.
 - ◆ Based on the **structured tables** in MIMIC III.
 - ◆ By leveraging the power of **crowd-sourcing**.
- **Develop a model for Question-to-SQL generation:**
 - ◆ Each question may be related to **multiple tables**.
 - ◆ **Related tables are unknown** for input questions.
 - ◆ Each question may be related to **multiple columns**.
 - ◆ The **dependencies** of different components are considered during the SQL generation.

Database Used: MIMIC III

- Medical Information Mart for Intensive Care III¹ (MIMIC III) dataset:
 - ◆ 46,520 de-identified ICU patients from Beth Israel Deaconess Medical Center between 2001 and 2012.
 - ◆ Different tables are generally linked by patient ID, hospital admission ID, lab item ID or ICD9 code.
 - ◆ Question answering on MIMIC III will involve five categories of patient information, including demographics, laboratory test results, diagnosis, procedures and prescriptions. We extracted and prepared a specific table for each category.

MIMICSQL Generation: Question Generation

- **Question generation:**

- ◆ Time-consuming to **manually** generate questions; Machine generated questions are not **natural**.
- ◆ We take advantage of **both template-based machine generation method and crowd-sourcing** to collect the question-SQL query pairs.

- **Template question:**

- ◆ The question templates are manually generated based on both the table **schema** and **content**. Two types of questions are included:
 - ◊ Information retrieval questions
 - ◊ Reasoning questions

- **Natural language (NL) question:**

- ◆ Each template question is **rephrased** as its corresponding natural language question by Freelancers with medical domain knowledge on a crowd-sourcing platform named **Freelancer**¹.

1. <https://www.freelancer.com/>

MIMICSQL Generation: SQL Generation

- A general SQL template:

```
SELECT AGG_OPERATION AGG_COLUMN  
FROM TABLE  
WHERE CONDITION
```

- ◆ **AGG_OPERATION:** the operation used for the corresponding selected AGG_COLUMN, takes one of the five values, including “NULL”, “COUNT”, “MAX”, “MIN” and “AVG”.
- ◆ **AGG_COLUMN:** the topic that we are interested in each question and it is based on table schema.
- ◆ **TABLE:** the table names that are related to the question.
- ◆ **CONDITION:** the constraints in the corresponding question. During the query generation, we mainly consider five different operations in the condition, including “=”, “>”, “<”, “>=” and “<=”.

Illustration of MIMICSQL

- MIMICSQL dataset is made publicly available at: <https://github.com/wangpinggl/TREQS>
- Here is an illustration example of MIMISQL:

Tables	DEMOGRAPHIC					PROCEDURES	PROCEDURES			
	SUBJECT_ID	HADM_ID	Gender	ADMISSION_TYPE	...		SUBJECT_ID	HADM_ID	SHORT_TITLE	...
	990	184231	F	EMERGENCY	...		9258	183354	Procedure-one vessel	...
	17772	122127	M	NEWBORN	...		28588	141664	Insert endotracheal tube	...
	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
	66411	178264	F	EMERGENCY	...		66411	178264	Abdomen artery incision	...
	29961	196409	M	EMERGENCY	...		66411	178264	Venous catch NEC	...

SQL template: SELECT \$AGG_OP (\$AGG_COLUMN)+ FROM \$TABLE WHERE (\$COND_COLUMN \$COND_OP \$COND_VAL)+

Question: How many female patients underwent the procedure of abdomen artery incision?

SQL query: SELECT COUNT (DISTINCT DEMOGRAPHIC.SUBJECT_ID)
FROM DEMOGRAPHIC INNER JOIN PROCEDURES on DEMOGRAPHIC.HADM_ID = PROCEDURES.HADM_ID
WHERE DEMOGRAPHIC."GENDER" = "F" AND PROCEDURES."SHORT_TITLE" = "Abdomen artery incision"

Data format: "sel": 1 , "agg_col": [[0, 0]] , "table": [0, 2], "cond": [[0, 6, 0, "F"], [2, 3, 0, "Abdomen artery incision"]]

Problem Formulation

Input:

Question:

How many patients whose primary disease is heart failure?

Semantic
Parsing
Method

Prediction:

Data format in MIMICSQL:

"sel":1, "agg_col":[[0,0]], "table":[0], "cond":[0, 13, 0], "heart failure"]

Output:

SQL:

SELECT COUNT DEMOGRAPHICS.SUBJECT_ID FROM DEMOGRAPHICS
WHERE DEMOGRAPHICS.DISEASE="heart failure";

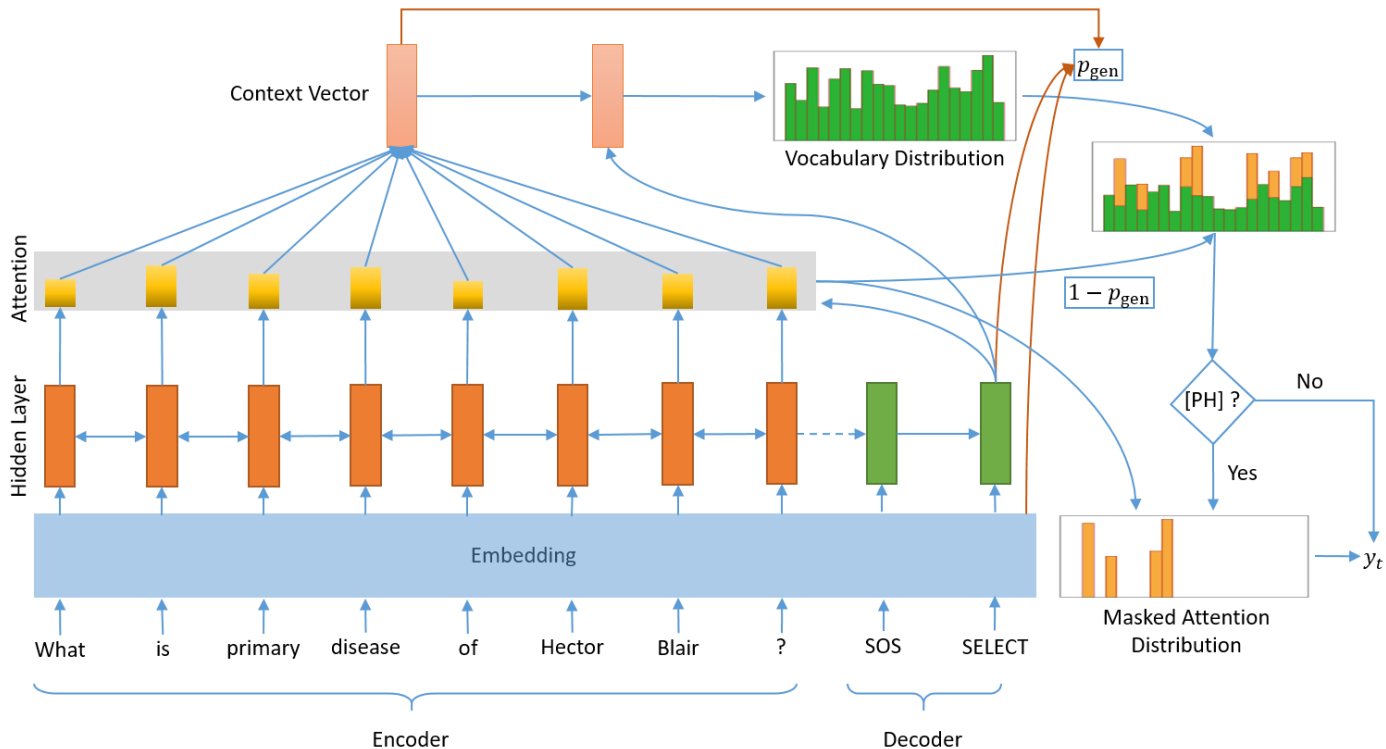
Our method

Language
Generation
Model

Translate-Edit Model for Question-to-SQL

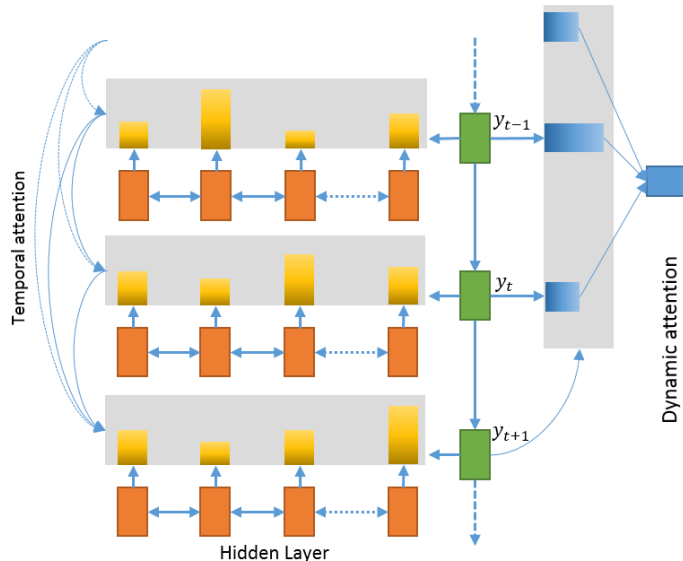
- Input: natural language question $x = (x_1, x_2, \dots, x_J)$.
- Output: SQL query $y = (y_1, y_2, \dots, y_T)$.
- **TR**anslate-**E**dit Model for **Q**uestion-to-**S**QL query (**TREQS**) generation:
 - ◆ **Translating** a question on the table contents to a SQL query using a Seq2Seq (Pointer-Generator) based model,
 - ◆ **Editing** the generated query with attentive-copying meta-algorithm.
 - ◆ **Further editing** the query with task-specific lookup tables.

Controlled Generation and Copying in TREQS



Attention Mechanisms in TREQS

- Temporal Attention on Question:
 - ◆ Prevent the decoder repetitively attending on the same part of the question.
- Dynamic Attention on SQL:
 - ◆ Dynamically attend on the previous generated tokens.



Condition Values Recovery

- Can generated queries be executable?
 - ◆ For example, “How many patients who have **bowel obstruct** and stay in hospital for more than 10 days?”
 - ◆ One of the conditions in the generated SQL query is “PRIMARY_DISEASE = bowel obstruct”, however, “bowel obstruct” does **not appear in the database**.
- Retrieve exact condition values based on the predicted ones and table content.
 - ◆ This approach makes use of **string matching metric ROUGE-L** to find the most similar condition value from the look-up table.

Evaluation

- **Overall evaluation:**

- ◆ **Execution accuracy:** $Acc_{ex} = \frac{N_{ex}}{N}$

- ◉ N denotes the number of questions in the collection.
- ◉ N_{ex} represents the number of executed queries that result in correct answers.

- ◆ **Logic form accuracy:** $Acc_{lf} = \frac{N_{lf}}{N}$.

- ◉ N_{lf} denotes the number of queries that match exactly with the ground truth.
- ◉ String match between generated queries and ground truth.

- **Break-down evaluation:**

- ◆ Accuracy for each component in SQL query.

Results of TREQS on Template Questions

Main results:

Method	Template Questions			
	Development		Testing	
	Acc_{LF}	Acc_{EX}	Acc_{LF}	Acc_{EX}
Coarse2Fine	0.298	0.321	0.518	0.526
M-SQLNET	0.258	0.588	0.382	0.603
Seq2Seq	0.098	0.372	0.160	0.323
Seq2Seq + recover	0.138	0.429	0.231	0.397
PtrGen	0.312	0.536	0.372	0.506
PtrGen + recover	0.442	0.645	0.426	0.554
TREQS (our)	<u>0.712</u>	<u>0.803</u>	<u>0.802</u>	<u>0.825</u>
TREQS + recover	0.853	0.924	0.912	0.940

Performance break-down:

Method	Development						Testing					
	Agg_{op}	Agg_{col}	$Table$	Con_{col+op}	Con_{val}	Average	Agg_{op}	Agg_{col}	$Table$	Con_{col+op}	Con_{val}	Average
Coarse2Fine	0.321	0.321	0.321	0.321	0.298	0.316	0.528	0.528	0.528	0.520	0.518	0.524
M-SQLNet	1.000	0.978	<u>0.994</u>	0.876	0.274	0.824	1.000	0.956	0.996	0.881	0.401	0.847
Seq2Seq	<u>0.999</u>	0.950	0.972	0.761	0.119	0.760	0.999	0.865	0.963	0.818	0.210	0.771
Seq2Seq + recover	<u>0.999</u>	0.950	0.972	0.761	0.163	0.769	0.999	0.865	0.963	0.818	0.296	0.788
PtrGen	<u>0.999</u>	<u>0.991</u>	0.992	0.979	0.325	0.857	1.000	0.988	<u>0.992</u>	0.985	0.381	0.869
PtrGen + recover	<u>0.999</u>	<u>0.991</u>	0.992	0.979	0.449	0.882	1.000	0.988	<u>0.992</u>	0.985	0.433	0.880
TREQS (our)	1.000	0.999	0.995	<u>0.924</u>	<u>0.719</u>	<u>0.927</u>	1.000	<u>0.995</u>	0.996	0.980	<u>0.810</u>	<u>0.956</u>
TREQS + recover	1.000	0.999	0.995	<u>0.924</u>	0.859	0.955	1.000	0.996	0.996	<u>0.984</u>	0.918	0.979

Results of TREQS on NL Questions

Main results:

Method	NL Questions			
	Development		Testing	
	Acc_{LF}	Acc_{EX}	Acc_{LF}	Acc_{EX}
Coarse2Fine	0.217	0.309	0.378	0.496
M-SQLNET	0.086	0.225	0.142	0.260
Seq2Seq	0.076	0.112	0.091	0.131
Seq2Seq + recover	0.092	0.195	0.103	0.173
PtrGen	0.126	0.174	0.160	0.222
PtrGen + recover	0.181	0.325	0.180	0.292
TREQS (our)	<u>0.451</u>	<u>0.511</u>	<u>0.486</u>	<u>0.556</u>
TREQS + recover	0.562	0.675	0.556	0.654

Performance break-down:

Method	Development						Testing					
	Agg_{op}	Agg_{col}	$Table$	Con_{col+op}	Con_{val}	Average	Agg_{op}	Agg_{col}	$Table$	Con_{col+op}	Con_{val}	Average
Coarse2Fine	0.319	0.313	0.321	0.260	0.214	0.285	0.524	0.490	0.528	0.448	0.413	0.481
M-SQLNet	0.994	0.939	0.933	0.722	0.080	0.734	<u>0.989</u>	0.873	0.941	0.749	0.140	0.738
Seq2Seq	0.978	0.872	0.926	0.466	0.137	0.676	0.970	0.696	0.892	0.563	0.239	0.672
Seq2Seq + recover	0.978	0.872	0.926	0.471	0.174	0.684	0.970	0.696	0.892	0.565	0.296	0.684
PtrGen	0.987	<u>0.917</u>	0.944	<u>0.795</u>	0.172	0.766	0.987	<u>0.830</u>	0.926	0.824	0.214	0.757
PtrGen + recover	0.987	<u>0.917</u>	0.944	<u>0.795</u>	0.236	0.776	0.987	<u>0.830</u>	0.926	0.824	0.235	0.760
TREQS (our)	<u>0.990</u>	0.912	<u>0.942</u>	0.834	0.574	0.850	0.993	0.827	0.941	<u>0.841</u>	<u>0.679</u>	<u>0.856</u>
TREQS + recover	<u>0.990</u>	0.912	<u>0.942</u>	0.834	0.694	0.873	0.993	0.827	0.941	0.844	0.763	0.874

Generated Queries on NL Questions

Method	Example 1	Example 2
Question	how many female patients underwent the procedure of abdomen artery incision?	how many patients admitted in emergency were tested for ferritin?
Ground truth	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "f" and procedures."short_title" = "abdomen artery incision"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_type" = "emergency" and lab."label" = "ferritin"
M-SQLNET	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "f" and procedures."short_title" = "parent infus nutrit sub"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_type" = "emergency" and lab."label" = "po2"
Seq2Seq	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "m" and procedures."long_title" = "other abdomen"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_location" = "phys referral/normal deli" and lab."itemid" = "ferritin"
Seq2Seq+recover	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "m" and procedures."long_title" = "other bronchoscopy"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_location" = "phys referral/normal deli" and lab."itemid" = "51200"
PtrGen	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "f" and procedures."long_title" = "spinal abdomen artery"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_type" = "emergency" and lab."label" = "troponin i"
PtrGen+recover	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "f" and procedures."long_title" = "spinal tap"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_type" = "emergency" and lab."label" = "troponin i"
TREQS	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "f" and procedures."short_title" = "abdomen artery abdomen"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_type" = "emergency" and lab."label" = "ferritin"
TREQS + recover	select count (distinct demographic."subject_id") from demographic inner join procedures on demographic.hadm_id = procedures.hadm_id where demographic."gender" = "f" and procedures."short_title" = "abdomen artery incision"	select count (distinct demographic."subject_id") from demographic inner join lab on demographic.hadm_id = lab.hadm_id where demographic."admission_type" = "emergency" and lab."label" = "ferritin"

* The incorrectly predicted words are highlighted in red color.

Visualization of Accumulated Attention on Conditions

Question	Generated SQL
provide the number of patients less than 83 years of age who were diagnosed with pneumococcal pneumonia .	select count (distinct demographic.subject_id) from demographic inner join diagnoses on demographic.hadm_id = diagnoses.hadm_id where demographic.age < 83 and diagnoses.short_title = pneumococcal pneumonia
provide the number of patients whose diagnoses is lap surg convert to open and they are female .	select count (distinct demographic.subject_id) from demographic inner join diagnoses on demographic.hadm_id = diagnoses.hadm_id where demographic.gender = f and diagnoses.long_title = lap surg convert to open they
among patients treated with amitriptyline , calculate the number of female patients .	select count (distinct demographic.subject_id) from demographic inner join prescriptions on demographic.hadm_id = prescriptions.hadm_id where demographic.gender = f and prescriptions.drug = amitriptyline
get me the number of elective hospital admission patients who had coronary artery primary disease .	select count (distinct demographic.subject_id) from demographic where demographic.admission_type = elective and demographic.diagnosis = coronary artery disease
give the number of patients whose admission type is elective and primary disease is abdominal abscess .	select count (distinct demographic.subject_id) from demographic where demographic.admission_type = elective and demographic.diagnosis = abdominal abscess
how many patients aged below 36 years have stayed in the hospital for more than 14 days ?	select count (distinct demographic.subject_id) from demographic where demographic.age < 36 and demographic.days_stay > 14
what is the number of patients whose admission location is emergency room admit and with primary disease t5 fracture ?	select count (distinct demographic.subject_id) from demographic where demographic.admission_location = emergency room admit and demographic.diagnosis = t5 fracture

* Different conditions are labeled with different colors.

* An intense shade on a word indicates a higher attention weight, and vice versa.

Conclusion

- Automated query generation from a natural language question is a challenging problem in healthcare domain.
- We first **generated MIMICSQL dataset with Question-SQL pairs** specifically for the Question-to-SQL generation task in healthcare domain.
- We further **proposed a Seq2Seq based TRanslate-Edit Model** for Question-to-SQL query (TREQS) generation and solved the problem on MIMICSQL by first generating the targeted SQL directly and then editing with both attention mechanism and recover technique.
- The proposed model is able to **handle several challenges** that are unique to MIMICSQL, including multiple tables, OOV words and table-unaware assumption.

Thank You!

Link to MIMICSQL dataset and codes:

<https://github.com/wangpinggl/TREQS>

Feel free to send questions and suggestions to

ping@vt.edu

tshi@vt.edu

redy@cs.vt.edu