DrugEHRQA: A Question Answering Dataset on Structured and Unstructured Electronic Health Records For Medicine Related Queries

Anonymous ACL submission

Abstract

This paper develops the first question answering dataset (DrugEHRQA) containing question-answer pairs from both structured tables and unstructured notes from a publicly available Electronic Health Record (EHR). EHRs contain patient records, stored in structured tables as well as unstructured clinical notes. The information in structured and unstructured EHR records is not strictly disjoint: information may be duplicated, contradictory, or provide additional context between these sources. This presents a rich opportunity to study question answering (QA) models that combine reasoning over both structured and unstructured data. Additionally, we propose a novel methodology that automatically generates a large QA dataset by retrieving answers from both structured and unstructured EHR records. The automatically-generated dataset has medication-related queries, containing over 70,000 question-answer pairs. Our dataset is validated for both individual modalities using state-of-the-art QA models. In order to address the problem arising from complex, nested queries, this is the first time Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers (RAT-SQL) has been used for EHR data. Finally, we introduce a rule-based method to obtain multi-modal answers, combining the answers from the different modalities. Our goal is to provide a benchmark dataset for multi-modal QA systems, and to open up new avenues of research in improving question answering over EHR structured data by using context from unstructured clinical data.

1 Introduction

Electronic Health Records (EHRs) are digitized records of patients' medical history, which can be in either structured or unstructured form. Question answering over EHRs aid doctors in diagnosing better, while it helps patients to obtain answers to health-related queries. The structured relational database has multiple tables that store information about the patient's demographics, diagnoses, medications, lab tests along with their results. The unstructured data, on the other hand, are notes entered by clinicians that contain a detailed description of every patient's visit, their past medical history, their problem, symptoms and more. Thus, to benefit from both the modalities (structured and unstructured), there arises a need for a multi-modal QA dataset on EHRs.

We present DrugEHRQA, the first QA dataset which uses both the structured tables and the unstructured clinical notes of an EHR to answer questions. The answers from the clinical notes are used to support or provide evidence to the answers retrieved from the structured tables. The former gives better context to support the latter. Moreover, there can be cases where a guaranteed answer might not be available in the structured tables, due to missing data/relation. For example, if the question is: 'What medication is the patient with an admission ID of 105104 taking for Hypoxemia?' The MIMIC-III tables have no direct relation between medicines and problems. The tables: DIAGNOSES_ICD and D_ICD_DIAGNOSES of MIMIC-III can be used to verify if the patient with admission ID 105104 is suffering from Hypoxemia, and the PRESCRIPTIONS table of MIMIC-III can be used to fetch all the medicines prescribed to the patient, having an admission ID of 105104. However, the patient could have been prescribed medicines for non-Hypoxemia related conditions, which will be contained in the tables. So, in such cases the unstructured clinical notes can be used to identify the medicines from this list, since the information about the medicine for Hypoxemia is directly present in the clinical notes.

One reason for the lack of any pre-existing multi-modal EHRQA dataset is due to the tedious amount of time and effort that is required to annotate such a dataset. In this work, we introduce a novel method to automatically generate a template-based drug

QA (DrugEHRQA) dataset from the MIMIC-III database. DrugEHRQA contains the following 1) natural language questions, 2) its corresponding SQL Query that can be used to retrieve answers from the multi-relational MIMIC-III tables, 3) the answers from either or both the modalities, and 4) the 'best selected' multi-modal answer. DrugEHRQA contains 70,381 QA pairs that have been generated using nine different template types. We also generated three paraphrases for every natural language question template, and analyzed the effects of paraphrasing on the baseline models. DrugEHRQA was benchmarked against existing models like TREQS (Wang et al., 2020b), RAT-SQL (Wang et al., 2020a), BERT QA (Devlin et al., 2019) and ClinicalBERT (Alsentzer et al., 2019) to test the validity of the DrugEHRQA dataset for the individual modalities

The main contributions of this paper are as follows:

1. (a) Introduce DrugEHRQA[1], the first QA dataset on multi-modal EHRs, containing QA pairs from structured tables and unstructured clinical notes of MIMIC III.
   (b) The dataset contains natural language questions, its corresponding SQL query for querying multi-relational tables of MIMIC-III, the retrieved answer(s) from either one or both the modalities, and also the combined multi-modal answer.
2. Introduce a novel technique to automatically generate a template-based dataset, without the need for any tedious manual annotations.

The remainder of the paper is organized into 8 sections. Section 2 discusses existing related work, Section 3 describes the DrugEHRQA dataset generation, Section 4 presents the analysis of DrugEHRQA, Section 5 discusses the implementation of structured and unstructured baseline models on DrugEHRQA, Section 6 discusses the reproducibility and limitations of our work, Section 7 proposes the broader impact of our dataset in the EHR QA research community and discusses possible future work, and Section 8 concludes the work.

2 Related Work

QA in EHRs has been limited to QA over knowledge bases (Wang et al., 2021), EHR tables (Wang et al., 2020b; Raghavan et al., 2021) or clinical notes (Johnson et al., 2016; Pampari et al., 2018). emrQA (Pampari et al., 2018) and CliniQG4QA (Yue et al.) are QA datasets that utilize unstructured text of EHRs to generate QA datasets. The emrQA contains 1 million question-logical forms along with over 40,000 QA evidence pairs, extracted from clinical notes of five n2c2 challenge datasets [2]. CliniQG4QA on the other hand, contains 1287 annotated QA pairs on 36 discharge summaries from clinical notes of MIMIC-III. CliCR (Šuster and Daelemans, 2018) is another large medical QA dataset which is constructed from clinical case reports. It is used for reading comprehension in the healthcare domain. The reports used in CliCR are called proxy for electronic health records, since the clinical reports look very similar to the discharge summaries of EHR.

There are QA datasets that are generated using template-based method like MIMICSQL (Wang et al., 2020b) and emrKBQA (Raghavan et al., 2021) which utilize the structured EHR tables of MIMIC-III for QA. emrKBQA contains 940,000 questions, logical forms and answers which uses the structured records of MIMIC-III. Both emrKBQA and emrQA use semi-automated methods to retrieve the answers. The question templates and logical forms are generated by the physicians, followed by a slot-filling process and answers are retrieved from MIMIC-III KB (Johnson et al., 2016). On the contrary, our dataset - DrugEHRQA uses both structured tables and clinical notes containing elaborate details of MIMIC-III to generate the QA dataset. We use an automatic novel methodology to create the dataset (described in Section 3).

3 Dataset Generation

The dataset has been generated using a template-based method. The dataset (DrugEHRQA) contains over 70,000 natural language questions. Each line in DrugEHRQA consists of a natural language question, its corresponding SQL query to retrieve answers from the MIMIC-III tables, the retrieved answers from MIMIC-III tables and/or answers from clinical notes of MIMIC-III, and the selected multi-modal answer. As stated earlier, generating a multi-modal dataset is time-consuming mainly because the data must be manually annotated, which is a very tedious process. To overcome this, we introduce a novel strategy to automatically gen-

[1] GitHub link of the source code and link to the dataset is not provided for anonymity purpose

[2] https://www.i2b2.org/NLP/DataSets/

erate the dataset. The dataset generation framework of DrugEHRQA is illustrated in figure 1. The dataset generation process can be explained using five steps: (1) Annotation of question templates, (2) Extraction of drug based relations from n2c2 repository, (3) Answer extraction from MIMIC-III tables, (4) Paraphrasing Natural Language Questions (5) Selecting multi-modal answers. The following subsections explain in detail the five steps involved in automatic data generation.

3.1 Annotation of Question Templates

We have annotated nine natural-language (NL) medicine-related question templates along with their corresponding SQL query templates. Five out of the nine NL question templates are taken from the medicine related templates of emrQA (Pampari et al., 2018). The authors created the remaining question templates on their own. The question templates are designed in such a way that their information appears in both structured and unstructured MIMIC-III data. The questions in the templates cover topics such as *drug-dosage, drug strength, route, form of medicine, problems*. Table A4 in the Appendix section shows the nine templates that have been used in the process of data generation. Each SQL query template is categorized into various difficulty levels- “easy”, “medium”, “hard” and “very hard”. The difficulty level is assigned based on the complexity of the SQL query, which is determined by number of where conditions, the number of aggregation columns, presence/absence of aggregation operators, group by, order by, limit, number of tables, joins and nesting. For example, the SQL query template in the first row of the table A4 is “easy” since it just has one aggregation column and one where condition. But the SQL query template in the last row is nested, contains joins and has multiple where conditions. Hence, it is classified as “very hard”. In the following sections, we use the terms “drug problems” and “drug reasons” interchangeably. This is because the data in the dataset is annotated as “drug reasons”, but to provide contextual clarity we use “drug problems” in this paper.

3.2 Answer Retrieval from Unstructured Data

"The 2018 Adverse Drug Event (ADE) dataset and Medical Extraction Challenge dataset" (Henry et al., 2020) present in the n2c2 repository [3] contains annotations for 505 clinical notes of patients (from the MIMIC-III database), who had experienced ADE while they were admitted in the hospital. This dataset will be henceforth referred to as challenge dataset. We used the annotations from the challenge dataset to extract all the drug related attributes for the 505 discharge summaries of patients in the MIMIC-III database. We used six drug-related attributes, namely, Strength-Drug, Form-Drug, Route-Drug, Dosage-Drug, Frequency-Drug, and Reason-Drug, from the challenge dataset to generate QA pairs. We used each of these drug attributes and the medicine names to generate nine types of natural language question templates. For example, the annotation from Dosage-Drug for a certain admission ID is used to answer the question - "What is the dosage of |drug| prescribed to the patient with admission id = |hadm_id|?", where |hadm_id| refer to the admission ID of the patient. This is depicted in the figure 1. Table A1 lists the drug attributes with examples and its derived NL questions. The medicines, drug attributes and admission IDs of the 505 annotation files are slot-filled to replace the placeholders in the question templates to generate the question-answer pairs. For data licensing issues of n2c2 repository, we submitted this QA dataset on clinical notes of MIMIC-III on n2c2 repository.

3.3 Answer Extraction from MIMIC-III Tables

Extraction of answers from MIMIC-III tables is achieved by using the admission IDs, names of drugs and problems, utilized in the data generation process from unstructured data (Section 3.2), to fill up the slots for |hadm_id|, |drug| and |problem| in the NL and SQL Query templates (Section‘3.1). Slot filling process was used to generate the SQL queries that helped in retrieving answers from the MIMIC-III’s structured database (refer figure 1. The answer may or may not exist in the MIMIC-III tables for the questions corresponding to the different combination of 505 admission IDs and entities of drugs (or problems) obtained from the clinical notes, resulting in an empty answer for certain questions. Three MIMIC-III tables, namely, PRESCRIPTIONS, DIAGNOSES_ICD, and D_ICD_DIAGNOSES are used for data retrieval. The PRESCRIPTIONS table of MIMIC-III contains drug-related information, whereas the tables - DIAGNOSES_ICD and

[3] https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

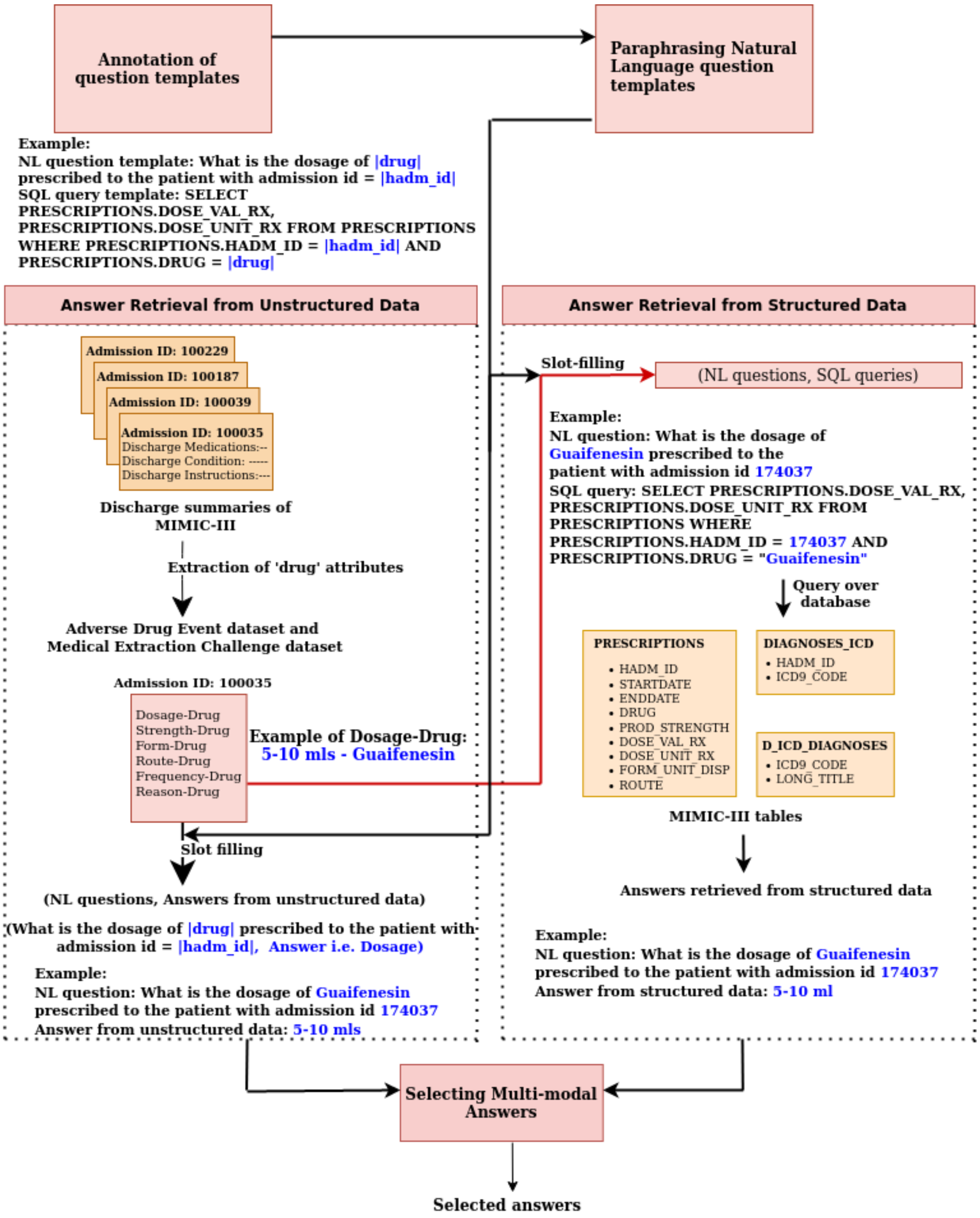

Annotation of question templates
Paraphrasing Natural Language question templates
Example:
NL question template: What is the dosage of |drug| prescribed to the patient with admission id = |hadm_id|
SQL query template: SELECT PRESCRIPTIONS.DOSE_VAL_RX, PRESCRIPTIONS.DOSE_UNIT_RX FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.HADM_ID = |hadm_id| AND PRESCRIPTIONS.DRUG = |drug|
Answer Retrieval from Unstructured Data
Admission ID: 100229
Admission ID: 100187
Admission ID: 100039
Admission ID: 100035
Discharge Medications:--
Discharge Condition: -----
Discharge Instructions:---
Discharge summaries of MIMIC-III
Extraction of 'drug' attributes
Adverse Drug Event dataset and Medical Extraction Challenge dataset
Admission ID: 100035
Dosage-Drug
Strength-Drug
Form-Drug
Route-Drug
Frequency-Drug
Reason-Drug
Example of Dosage-Drug: 5-10 mls - Guaifenesin
Slot filling
(NL questions, Answers from unstructured data)
(What is the dosage of |drug| prescribed to the patient with admission id = |hadm_id|, Answer i.e. Dosage)
Example:
NL question: What is the dosage of Guaifenesin prescribed to the patient with admission id 174037
Answer from unstructured data: 5-10 mls
Answer Retrieval from Structured Data
Slot-filling
(NL questions, SQL queries)
Example:
NL question: What is the dosage of Guaifenesin prescribed to the patient with admission id 174037
SQL query: SELECT PRESCRIPTIONS.DOSE_VAL_RX, PRESCRIPTIONS.DOSE_UNIT_RX FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.HADM_ID = 174037 AND PRESCRIPTIONS.DRUG = "Guaifenesin"
Query over database
PRESCRIPTIONS
HADM_ID
STARTDATE
ENDDATE
DRUG
PROD_STRENGTH
DOSE_VAL_RX
DOSE_UNIT_RX
FORM_UNIT_DISP
ROUTE
DIAGNOSES_ICD
HADM_ID
ICD9_CODE
D_ICD_DIAGNOSES
ICD9_CODE
LONG_TITLE
MIMIC-III tables
Answers retrieved from structured data
Example:
NL question: What is the dosage of Guaifenesin prescribed to the patient with admission id 174037
Answer from structured data: 5-10 ml
Selecting Multi-modal Answers
Selected answers

Figure 1: Dataset generation framework of DrugEHRQA. There are five steps in this process: (1) annotation of question templates, (2) answer retrieval from unstructured clinical notes, (3) answer retrieval from structured EHR Data, (4) paraphrasing natural language question templates, and (5) selecting multi-modal answers.

D_ICD_DIAGNOSES contain the diagnosed results of the patients. The DrugEHRQA dataset now contains NL Questions, its corresponding SQL queries for querying the structured database, the answers retrieved from the structured tables (Answer_Structured), and the answers retrieved from unstructured data (Answer_Unstructured).

3.4 Paraphrasing Natural Language Questions

Patients and clinicians may pose the same question in different formats (paraphrases). There has been a substantial amount of work done in EHR QA, studying the effects of NL paraphrasing in QA (Wang et al., 2020b; Pampari et al., 2018; Rawat et al., 2020; Soni and Roberts, 2019; Moon and Fan, 2020). We added paraphrases in the natural language question templates to improve the diversity of DrugEHRQA dataset, making it more realistic, and more robust. We created four paraphrases for each of the nine natural language query templates (i.e. three additional paraphrases per template). The figure 2 depicts an example of paraphrasing an NL question template. The SQL queries are randomly mapped to one of the four paraphrased NL questions.

3.5 Selecting Multi-modal Answers

Whenever a patient is admitted to the hospital, all their treatment and medication details are immediately stored in the EHR tables (i.e. they are up-to-date). The clinical notes have elaborate details but may have outdated records, and hence less-accurate. Hence, between the two modalities, the structured records can be considered as a more authentic source of information. Therefore, in most cases the answers retrieved from structured records are considered more precise than the answers from unstructured data. This is especially true when answers directly exist in the MIMIC-III tables (i.e. non-derived relation queries). In DrugEHRQA, questions concerning: (a) Dosage of medicine prescribed to the patient, (b) Route of medicine, (c) Form of medicine, and (d) List of medicines prescribed to the patient are some examples where answers exist directly in the MIMIC-III tables.

There are certain queries in DrugEHRQA, for which a direct answer is not available in the MIMIC-III tables (i.e. derived relation queries) because of missing data/relations. Let's consider using MIMIC III tables to answer the question: 'What medication is the patient with an admission ID of 105104 taking for Hypoxemia?'. MIMIC-III tables contain information about the patient of interest being diagnosed with 'Hypoxemia'. They also contain the list of medicines prescribed to the patient of interest. However, the tables may contain records (medicines) prescribed to the patient for non-Hypoxemia related conditions. In this scenario, the answer from unstructured data for such missing relations is more reliable since the answer is directly available in the clinical notes.

We have used a two-step process to generate the multi-modal answers. In the first step, an automatic method was used to retrieve the multi-modal answer.

To automatically generate the multi-modal answers, we follow three major rules. Table A2 helps to explain the rules below using examples.

- If the answer exists in only one modality, the available answer is selected as the multi-modal answer. (1st row, Table A2).

- Check for overlapping answers. If there is even one common answer between Answer_Structured and Answer_Unstructured, choose the common answer. (2nd row, Table A2).

- If there are no common answers between the two modalities, choose the answer from the modality which is more reliable. (4th row, Table A2). In the last row of Table A2, we can observe that the answers from the two modalities are different. Since the question is a non-derived relation query, the answer from the structured database is selected as the multi-modal answer.

After generating the multi-modal answers automatically, the author manually sampled 500 queries, and cross-checked the results for the multi-modal answer. Please refer to the supplementary materials for further details regarding the human validation process.

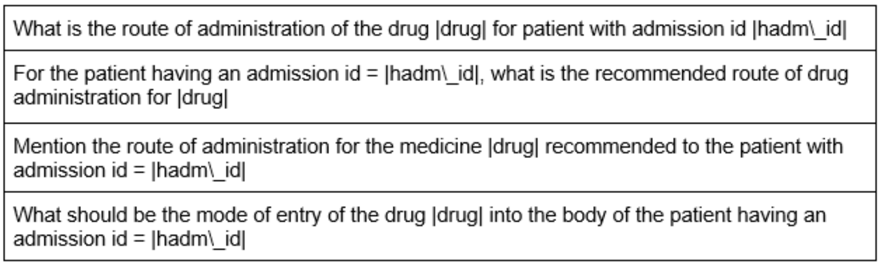
4 Analysis of the DrugEHRQA dataset

The SQL queries generated in the DrugEHRQA dataset can be classified into easy, medium, hard and very hard SQL queries (Refer Table 1). The generated SQL queries were classified using the complexity determination method used in RAT-SQL (Wang et al., 2020a). Complexities of the SQL queries are determined by factors like number of tables in the SQL query, number of conditions, presence of nesting etc. The DrugEHRQA dataset contains more complicated SQL queries (containing nested queries) than the existing text to SQL datasets in EHR like MIMICSQL (Wang et al., 2020b)

Table 1: Complexity levels of SQL queries in the DrugEHRQA dataset

Difficulty levels	Percentage of queries
Easy	1.1%
Medium	39.2%
Hard	9.8%
Very Hard	49.9%

The DrugEHRQA contains a total of 70,381 questions along with answers from either the multi-relational tables, or the unstructured clinical data of MIMIC-III, or from both the sources. The dataset also contains an automatically generated multi-modal answer. Roughly 41 % of the drug-related queries can be answered individually by the structured data and unstructured data. There are a total of 12,738 samples, which is approximately



What is the route of administration of the drug |drug| for patient with admission id |hadm_id|
For the patient having an admission id = |hadm_id|, what is the recommended route of drug administration for |drug|
Mention the route of administration for the medicine |drug| recommended to the patient with admission id = |hadm_id|
What should be the mode of entry of the drug |drug| into the body of the patient having an admission id = |hadm_id|

Figure 2: Example of various paraphrases of a natural language question template in the DrugEHRQA dataset.

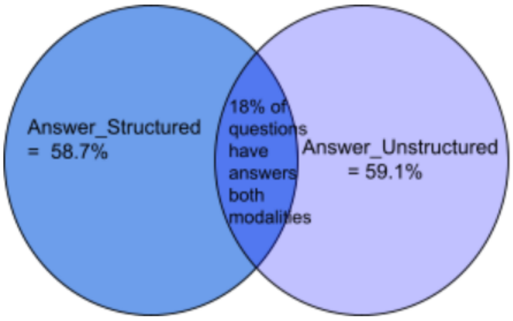

Answer_Structured = 58.7%
18% of questions have answers both modalities
Answer_Unstructured = 59.1%

Figure 3: Venn diagram showing percentage of answers available in structured and unstructured records

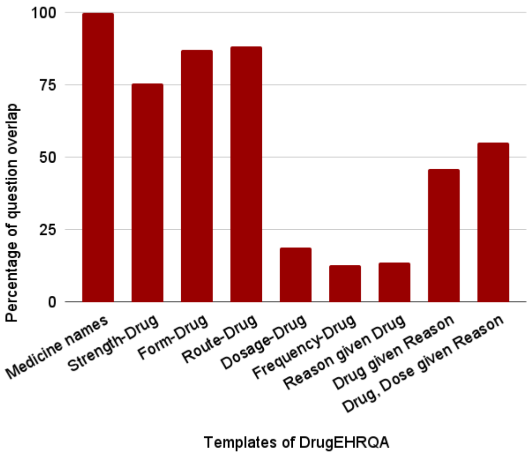


Figure 4: Percentage of questions with at least one answer-overlap from text-table QA

18% of the total questions that contain answers in both tables and text (shown in Figure 3). Also, out of the 12,738 queries containing answers in both structured and unstructured EHR data, 15% of this dataset have missing relations (or information) in the structured tables. Hence, among the queries containing answers in both the modalities, the answers from unstructured EHR data is more reliable (than structured EHR data) for 15% of the queries.

We analyzed the 18% of the samples (Figure 3) containing answers in both the modalities, and concluded that the information in structured and unstructured EHR records is not strictly disjoint - information may be duplicated, contradictory, or provide additional context between these sources. Figure 4 shows the percentage of questions with at least one-answer overlap between table and text QA for the nine templates. Some of the templates like medicine names, form-drug and route-drug have a high percentage of overlapping answers. Confidence in the accuracy of answers increases when the answers are the same, e.g.: row 2 and row 3 of Table A2. Table A3 shows examples where multi-modal QA in EHRs can help provide additional context. We observed that the answers from the modalities were different, but the dual modalities together provide the complete answer. The answer from structured data gives the dosage in milligrams, whereas the answer retrieved from the clinical notes presents the dosage based on the number of tablets. Both of the answers are right, which can be verified from the last column, since the dosage recommended in row 1 is one 325 mg tablet, to be taken daily. In short, answers from one modality can help to provide better context to the answers retrieved from the other modality.

5 Baseline models for QA over EHRs

This section discusses all the baseline models that we used for performing QA tasks on our dataset. We use separate QA baseline models to validate our QA dataset on structured EHRs and unstructured EHRs. Two existing models - TREQS (Wang et al., 2020b) and RAT-SQL (Wang et al., 2020a) are used for text-to-SQL tasks on DrugEHRQA using MIMIC-III tables.

TRanslate-Edit Model for Question-to-SQL (TREQS) (Wang et al., 2020b) is a sequence-to-sequence model which generates SQL query for a given question. It also makes the necessary modifications with the help of an attentive copying mechanism and task-specific look-up tables. TREQS was unable to handle text-SQL pairs when the SQL queries were nested (for 4 out of 9 templates), so we had to use RAT-SQL (Relation-

Aware Schema Encoding and Linking for Text-to-SQL Parsers) (Wang et al., 2020a) to test the remaining templates. This is the first time RAT-SQL is being introduced in the healthcare domain. MIMICSQL dataset (Wang et al., 2020b) has relatively simple queries, so using TREQS model was sufficient. In order to address more complex SQL queries of the DrugEHRQA dataset, we had to use a more advanced text-to-sql model. RAT-SQL uses a relation-aware self-attention mechanism to address schema encoding, schema linking, and feature representation within a text-to-SQL encoder. Self-aware attention mechanism in RAT-SQL helps to encode more complex relationships between columns and tables within the schema of the database, as well as between the question and the database schema.

BERT QA (Devlin et al., 2019) and Clinical-BERT QA (Alsentzer et al., 2019) has gained popularity over the years for QA over unstructured data (Johnson et al., 2016; Soni and Roberts, 2020). ClinicalBERT is the clinical version of BERT pre-trained on the clinical notes of MIMIC-III. The BERT QA model is pre-trained on large datasets like BooksCorpus and English Wikipedia. The training size of Clinical BERT's corpus (roughly 50M words) is much smaller than BERT (roughly 3300M words).

5.1 Experimental Setup

We used a sample dataset of 10,787 text-SQL pairs, 12,737 text-SQL pairs and 12,508 QA pairs for TREQS, RAT-SQL and BERT/ClinicalBERT respectively, and for all our experiments we split the dataset in the ratio of 0.8/0.1/0.1 to obtain train, dev and test sets, and trained the model with a batch size of 16, 20, and 12 respectively. The difference in number of samples between TREQS, RAT-SQL, and BERT/ClinicalBERT is due to the limitations of the model in only supporting 5, 9, and 8 (out of the 9 templates) respectively. We used a smaller sample of the dataset for our experiments for resource constraints.

We trained the TREQS model for 4 epochs with a learning rate of 0.005, grad clip of 2.0, and a maximum vocabulary size of 50,000. For the scheduler, we used a step size of 2 and step decay of 0.8 and set the minimum word frequency to 5. The model was trained on an Intel i7 (8th gen) with hyperthreading enabled and 32 GB RAM. For the RAT-SQL model, we used GloVe (Pennington et al., 2014) word embeddings for the 50 most commonly occurring words in the training data. The model was trained using GeForce RTX 2080 Ti up to 40,000 steps while using Adam optimizer (Kingma and Ba, 2015). The same hyperparameters were used as stated in (Wang et al., 2020a). For BERT and ClinicalBERT, Quadro RTX 6000 GPU was used for training the model for 2 epochs with a learning rate of 3e-5. A doc stride of 128 is used with a maximum sequence length of 384. Before fine-tuning BERT and Clinical-BERT on DrugEHRQA dataset, the pre-trained models of BERT and Clinical BERT are fine-tuned on SQUAD (Rajpurkar et al., 2016).

5.2 Results of QA of DrugEHRQA on structured EHR tables

We use Logical Form Accuracy (Acc_LF) and Execution Accuracy (Acc_Ex) as evaluation metrics to test the SQL queries for the TREQS model. Logical Form Accuracy can be defined as the ratio of the number of strings matched between the ground truth and the generated SQL query, to the total number of question-SQL pairs. Execution Accuracy on the other hand, represents the ratio of the number of SQL queries generated with correct answers to the total number of question-SQL pairs. Table 2 shows the overall performance of the TREQS model, while predicting SQL queries from the NL questions in the test set. At times, the condition value in the question may not match the table's header. The TREQS model uses a recover technique where a string matching metric, ROUGE-L, is used to search for the most similar condition value using the lookup table for every predicted SQL query. Hence, the "TREQS (with recover)" in Table 2 refers to the accuracy of the test set when the query generated using the sequence-to-sequence model is further edited to recover the exact data with the help of the table schema and look-up tables of content keywords. We observe from the table that after using recover, the overall performance improves.

Table 2 displays the overall accuracy of DrugEHRQA on the RAT-SQL model. As expected, the logical form accuracy of the predicted SQL queries is slightly lesser for paraphrased DrugEHRQA than non-paraphrased DrugEHRQA. Also, we observe that the overall LF accuracy of DrugEHRQA on RAT-SQL is much higher than the TREQS model. This is because the computation of

Table 2: Overall performance of DrugEHRQA on TREQS and RAT-SQL models

Models	Acc_LF	Acc_EX
TREQS (without recover)	0.618	0.618
TREQS (with recover)	0.623	0.624
RAT-SQL	0.8723	-

Table 3: Results of QA using BERT and Clinical BERT on clinical notes of DrugEHRQA

	Dev Exact-match	Dev F1-score	Test Exact-match	Test F1-score
BERT	79.806	83.266	80.158	83.561
Clinical BERT	79.725	82.801	80.238	83.289

LF accuracy in RAT-SQL evaluates the predicted SQL query on all components except the condition values of the SQL query. Prediction of the condition values in a text-to-SQL prediction task is much more challenging than predicting the other components of the SQL Query. The section A in the appendix section compares the performance of DrugEHRQA dataset with the existing datasets on the different QA models.

5.3 Results of QA of DrugEHRQA on unstructured EHR data

We evaluate our dataset with exact match and F1 score as evaluation metrics. Our dataset performs fairly well on the test set with an exact score of 80.158 and an F1 score of 83.289 for BERT QA (Table 3). We obtain a marginal difference in performance between BERT and ClinicalBERT.

6 Reproducibility & Limitations

The user must have credentialed access to PhysioNet[4]. The user must download the MIMIC-III data, retrieve the drug relations from the '2018 (Track 2) Adverse Drug Event (ADE) and the Medication Extraction Challenge dataset' from the n2c2 repository (after requesting for access to n2c2 datasets). Once this is done, the user can just replicate the steps described in the dataset generation process (Section 3) to produce the DrugEHRQA dataset. Even though the multimodal QA dataset generation process is automatic, without the need for long hours of annotation. But this procedure is limited only to the MIMIC-III database. The same steps cannot be reproduced for other EHR databases. In fact, MIMIC-IV (Johnson A, 2020) is the latest version. But since the dataset generation process is dependent on the drug relations extracted from the '2018 (Track 2) Adverse Drug Event (ADE) and the Medication Extraction Challenge dataset', so our dataset generation process was limited to the MIMIC-III database.

7 Broader Impact on the EHR QA research community and Future Work

The DrugEHRQA dataset helps to put a spotlight on multimodal EHRs. The data in the structured and unstructured EHR may contain duplicated information (improves confidence of the answer), they may contrast each other, and may also aid in adding context to each other. This opens up new avenues of research in multimodal QA in EHRs. DrugEHRQA can be used as a benchmark model for all QA models that uses multiple EHR tables and clinical notes for information retrieval. Since in a lot of cases, the data in structured and unstructured EHR sources helps to provide additional context to each other, another possible application of DrugEHRQA is in improving QA over structured (or unstructured), by using information or evidence from the unstructured EHR source (or structured).

8 Conclusion

To conclude, EHRs contain a large amount of up-to-date patient information in the structured databases, along with clinical notes containing elaborate details. We have introduced a novel methodology to generate a large multimodal QA dataset, containing answers from multi-relational tables and discharge summaries of a publicly available EHR database (MIMIC-III). It is the first QA dataset which contains natural language questions, SQL queries, and answers from either or both structured EHR tables and unstructured free text. Additionally, we use an automated methodology to generate the multimodal answer. Following this, human annotators verified the answers for a sampled dataset. To validate our dataset, we have used existing state-of-the-art models for QA over structured EHRs, as well as QA over unstructured EHRs. This dataset introduces new horizons of research in multimodal QA over EHRs.

[4] https://physionet.org/

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner. 2020. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association*, 27(1):3–12.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Pollard T Horng S Celi L A Mark R. Johnson A, Bulgarelli L. 2020. Mimic-iv (version 0.4). https://doi.org/10.13026/a3wn-hq05.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Sungrim Riea Moon and Jungwei Fan. 2020. How you ask matters: The effect of paraphrastic questions to bert performance on a clinical squad dataset. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 111–116.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Preethi Raghavan, Jennifer J Liang, Diwakar Mahajan, Rachita Chandra, and Peter Szolovits. 2021. emrkbqa: A clinical knowledge-base question answering dataset. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 64–73.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Bhanu Pratap Singh Rawat, Wei-Hung Weng, So Yeon Min, Preethi Raghavan, and Peter Szolovits. 2020. Entity-enriched neural models for clinical question answering. In *Proceedings of the 19th SIG-BioMed Workshop on Biomedical Language Processing*, pages 112–122.

Sarvesh Soni and Kirk Roberts. 2019. A paraphrase generation system for ehr question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 20–29.

Sarvesh Soni and Kirk Roberts. 2020. Evaluation of dataset selection for pre-training and fine-tuning transformer language models for clinical question answering. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5532–5538.

Simon Šuster and Walter Daelemans. 2018. Clicr: A dataset of clinical case reports for machine reading comprehension. In *Proceedings of NAACL-HLT*, pages 1551–1563.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020a. 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.

Ping Wang, Tian Shi, Khushbu Agarwal, Sutanay Choudhury, and Chandan K Reddy. 2021. Attention-based aspect reasoning for knowledge base question answering on clinical notes. *arXiv e-prints*, pages arXiv–2108.

Ping Wang, Tian Shi, and Chandan K Reddy. 2020b. Text-to-sql generation for question answering on electronic medical records. In *Proceedings of The Web Conference 2020*, pages 350–361.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921.

Xiang Yue, Xinliang Frederick Zhang, Ziyu Yao, Simon Lin, and Huan Sun. Cliniqg4qa: Generating diverse questions for domain adaptation of clinical question answering.

Appendix

A Performance comparison of DrugEHRQA with existing datasets on different QA models

Figure A3a shows the performance of TREQS on DrugEHRQA, comparing it with its performance on MIMICSQL dataset. It can be observed from the table that the LF accuracy and execution accuracy of TREQS on MIMICSQL is lower than DrugEHRQA. This is because the queries in DrugEHRQA dataset are much more complex than the queries in the MIMICSQL dataset. We also observe from Figure A3a that after adding paraphrases to our dataset, the accuracy of the model decreased by a very small amount compared to the non-paraphrased DrugEHRQA dataset. This is because paraphrasing of natural language questions in the dataset increases complexity in the NL question to SQL task.

We compare the exact match accuracy of RAT-SQL on DrugEHRQA with MIMICSQL and Spider dataset (Yu et al., 2018)(See Figure A3b). The exact match accuracy of RAT-SQL model on DrugEHRQA is higher than its exact match accuracy in the Spider dataset. This is because the Spider dataset makes use of multiple databases unlike DrugEHRQA, thus making their task of text-to-sql prediction more challenging. But since the SQL queries predicted in DrugEHRQA are much more difficult in comparison to MIMICSQL, the exact match accuracy of DrugEHRQA on RAT-SQL is slightly lesser than in MIMICSQL dataset.

Figure A1 shows the performance of our dataset compared to emrQA for QA on BERT and ClinicalBERT. We have used only the factoid questions of emrQA for evaluation. The DrugEHRQA performs much better than emrQA. Figure A2 shows comparison in performance of DrugEHRQA on ClinicalBERT when the NL questions have been paraphrased, versus when they are not paraphrased. From the table, we observe that there is a significant decline in exact match and F1 score after paraphrasing, if the model has not been fine-tuned on SQUAD (Rajpurkar et al., 2016). But after fine-tuning on SQUAD, the difference in their performance is negligible.

B Question templates with examples

This section uses Table A1, table A2, table A3, and table A4 to list the different question templates, followed by some examples. The Table A1 describes the different question templates of the dataset derived from the drug attributes and entities in the "2018 (Track 2) Adverse Drug Event (ADE) and the Medication Extraction Challenge dataset". Table A2 and Table A3 displays examples from the dataset where the two modalities (i.e. structured and unstructured EHR data) contain similar answers (for example, 2nd and 3rd row of table A2), when the two modalities contain conflicting or dissimilar answers (example: 4th row of table A2), and also shows examples where the answers retrieved from structured and unstructured EHR data complement each other (for example, row 1 and 2 of table A3). The rules described in Section 3.5 was used to obtain the multimodal answers. Finally, the table A4 lists the NL question templates, its corresponding SQL query templates, and their difficulty level.

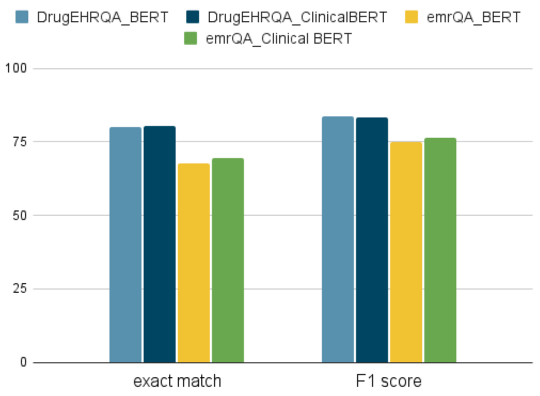


Figure A1: Comparing performance of DrugEHRQA with emrQA after fine-tuning on SQUAD.

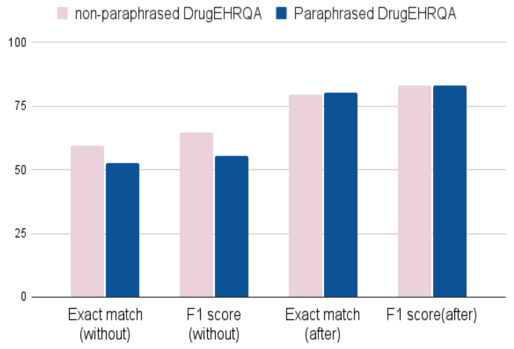
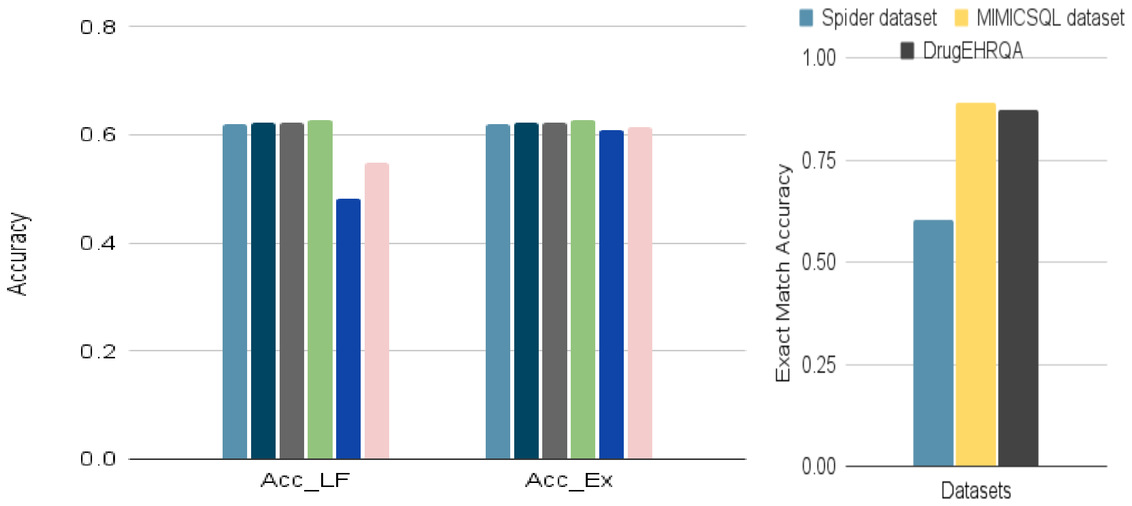


Figure A2: Performance comparison of paraphrased DrugEHRQA with non-paraphrased DrugEHRQA on Clinical BERT. Note: "(Without)" refers to directly fine-tuning on DrugEHRQA and "(after)" refers to fine-tuning on SQUAD before fine-tuning our dataset.



(a)

(b)

Figure A3: (a) Overall Accuracy of TREQS model on DrugEHRQA dataset and MIMICSQL dataset. (b) Exact Match Accuracy of RAT-SQL model on DrugEHRQA, Spider, and MIMICSQL dataset.

Table A1: NL Question templates derived from drug-related entities and attributes extracted from the clinical notes using the n2c2 dataset, along with examples

Drug attributes and entities	Examples	NL Question templates
Drug	Lithium Carbonate, Propafenone	What are the list of medicines prescribed to the patient
Strength-Drug	(300mg, Lithium Carbonate)	What is the drug strength of |drug|
Form-Drug	(Tablet, Propafenone)	What is the form of |drug|
Route-Drug	(PO, Metoprolol Tartrate)	What is the route of administration for the drug |drug|
Dosage-Drug	(One tablet, Bactrim)	What is the dosage of |drug| prescribed to the patient
Frequency-Drug	(14 day, Zosyn)	How long has the patient been taking |drug|
Reason-Drug	(Constipation, Polyethylene Glycol)	Why is the patient been given |drug|
Reason-Drug	(Polyethylene Glycol, Constipation)	What is the medication prescribed to the patient for |problem|
Reason-Drug, Dosage-Drug	(Constipation, Polyethylene Glycol), (300mg, Polyethylene Glycol)	List all the medicines and their dosages prescribed to the patient for |problem|

Table A2: Rules for automatic multi-modal answer retrieval

Question	Answer-_Structured	Answer-_Unstructured	Multi-modal answer
WHAT IS THE MEDICATION PRESCRIBED TO THE PATIENT WITH ADMISSION ID 111160 FOR PAIN	–	MORPHINE	MORPHINE
WHAT IS THE DRUG STRENGTH OF SIMETHICONE PRESCRIBED TO THE PATIENT WITH ADMISSION ID 125206	80MG TABLET	80 MG	80MG TABLET
HOW LONG HAS THE PATIENT WITH ADMISSION ID = 187782 BEEN TAKING VANCOMYCIN	14 DAYS	14 DAYS	14 DAYS
WHAT IS THE DRUG STRENGTH OF FUROSEMIDE PRESCRIBED TO THE PATIENT WITH ADMISSION ID 100509	40MG/4ML VIAL	10 MG	40MG/4ML VIAL

Table A3: Information in structured and unstructured EHR providing additional context to each other. Note that the field 'Answer_Unstructured' is the direct answer extracted from unstructured data with the help of the n2c2 dataset, and the field 'Phrases from clinical notes' are the lines of text in the discharge summary from which the answer is extracted.

NL Questions	Answer-_Structured	Answer-_Unstructured	Phrases from clinical notes
WHAT IS THE DOSE OF ASPIRIN THAT THE PATIENT WITH ADMISSION ID = 142444 HAS BEEN PRESCRIBED	325MG,300MG	ONE (1)	325 mg Tablet Sig: One (1) Tablet PO DAILY (Daily). 5. Acetaminophen 325 mg Tablet Sig: One (1) Tablet PO Q6H (every 6 hours) as needed.
LIST ALL THE MEDICINES AND THEIR DOSAGES PRESCRIBED TO THE PATIENT WITH ADMISSION ID = 105014 FOR POLYMYALGIA RHEUMATICA	PREDNISONE: 20 MG, TACROLIMUS: 4 MG, MYCOPHENOLATE MOFETIL: 1000 MG, TACROLIMUS: 4 MG, TACROLIMUS: 5 MG, MYCOPHENOLATE MOFETIL: 500 MG	PREDNISONE: ONE (1)	20 mg Tablet Sig: One (1) Tablet PO DAILY (Daily).

Table A4: Templates and their level of difficulty

Sl. No	NL Question Template	SQL Query Template	Difficulty Level
1.	What are the list of medicines prescribed to the patient with admission id |hadm_id|	SELECT PRESCRIPTIONS.DRUG FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.HADM_ID = |hadm_id|	Easy
2.	What is the drug strength of |drug| prescribed to patient with admission id |hadm_id|	SELECT PRESCRIPTIONS.PROD_STRENGTH FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.HADM_ID = |hadm_id| AND PRESCRIPTIONS.DRUG = |drug|	Medium
3.	What is the form of |drug| prescribed to patient with admission id |hadm_id|	SELECT PRESCRIPTIONS.FORM_UNIT_DISP FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.DRUG = |drug| AND PRESCRIPTIONS.HADM_ID = |hadm_id|	Medium
4.	What is the route of administration for the drug |drug| for patients with admission id = |hadm_id|	SELECT PRESCRIPTIONS.ROUTE FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.DRUG = |drug| AND PRESCRIPTIONS.HADM_ID = |hadm_id|	Medium
5.	What is the dosage of |drug| prescribed to the patient with admission id = |hadm_id|	SELECT PRESCRIPTIONS.DOSE_VAL_RX, PRESCRIPTIONS.DOSE_UNIT_RX FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.HADM_ID = |hadm_id| AND PRESCRIPTIONS.DRUG = |drug|	Medium
6.	How long has the patient with admission id = |hadm_id| been taking |drug|	SELECT SUM(PRESCRIPTIONS.DURATION_IN_DAYS) FROM PRESCRIPTIONS WHERE PRESCRIPTIONS.HADM_ID = |hadm_id| AND PRESCRIPTIONS.DRUG = |drug| GROUP BY PRESCRIPTIONS.HADM_ID, PRESCRIPTIONS.DRUG	Hard
7.	Why is the patient with admission id = |hadm_id| been given |drug|	SELECT L3.SHORT_TITLE FROM D_ICD_DIAGNOSES AS L3 WHERE L3.ICD9_CODE IN (SELECT L1.ICD9_CODE FROM DIAGNOSES_ICD AS L1 INNER JOIN PRESCRIPTIONS AS L2 ON L1.HADM_ID = L2.HADM_ID WHERE L1.HADM_ID = |hadm_id| AND L2.DRUG = |drug|)	Very hard

8.	What is the medication prescribed to the patient with admission id = |hadm_id| for |problem|	SELECT Y.DRUG FROM PRESCRIPTIONS AS Y WHERE Y.HADM_ID = (SELECT L1.HADM_ID FROM DIAGNOSES_ICD AS L1 INNER JOIN D_ICD_DIAGNOSES AS L2 ON L1.ICD9_CODE = L2.ICD9_CODE WHERE L1.HADM_ID = |hadm_id| AND L2.LONG_TITLE = |problem|)	Very hard
9.	List all the medicines and their dosages prescribed to the patient with admission id = |hadm_id| for |problem|	SELECT Y.DRUG, Y.DOSE_VAL_RX, Y.DOSE_UNIT_RX FROM PRESCRIPTIONS AS Y WHERE Y.HADM_ID = (SELECT L1.HADM_ID FROM DIAGNOSES_ICD AS L1 INNER JOIN D_ICD_DIAGNOSES AS L2 ON L1.ICD9_CODE = L2.ICD9_CODE WHERE L1.HADM_ID = |hadm_id| AND L2.LONG_TITLE = |problem|)	Very hard