emrKBQA: A Clinical Knowledge-Base Question Answering Dataset

Preethi Raghavan[1,3,*], **Diwakar Mahajan**[1,3,#], **Jennifer Liang**[1,3,§],
Rachita Chandra[1,3,†], **Peter Szolovits**[2,3,‡]
[1]IBM Research, [2]MIT CSAIL, [3]MIT-IBM Watson AI Lab
{*praghav,#dmahaja,§jjliang,†rachitac}@us.ibm.com,‡psz@mit.edu

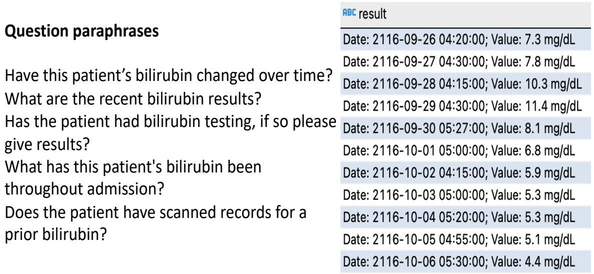
Abstract

We present emrKBQA, a dataset for answering physician questions from a structured patient record. It consists of questions, logical forms and answers. The questions and logical forms are generated based on real-world physician questions and are slot-filled and answered from patients in the MIMIC-III KB (Johnson et al., 2016) through a semi-automated process. This community-shared release consists of over 940000 question, logical form and answer triplets with 389 types of questions and ≈7.5 paraphrases per question type. We perform experiments to validate the quality of the dataset and set benchmarks for question to logical form learning that helps answer questions on this dataset.

1 Introduction

The last decade has seen widespread adoption of electronic health records (EHRs) across hospitals and clinics in the US (Jha et al., 2006; Evans, 2016). Physicians often seek answers to questions from a patient's EHR to support clinical decision-making (Demner-Fushman et al., 2009). It is not too hard to imagine a future where a physician interacts with an EHR system and asks it complex questions and expects precise answers, with adequate context, from a patient's record (Pampari et al., 2018). Central to such a world is a medical question answering system that processes natural language questions asked by physicians and finds answers to the questions in structured and unstructured sources in the patient's record.

However, the longitudinal, domain specific nature of patient records along with privacy concerns makes it difficult to develop large-scale annotated datasets for training machine learning models. This motivated Pampari et al. (2018) to develop the first community-shared patient QA dataset, emrQA, using a semi-automated process and create a large-scale dataset with over 1M question-answer and question-logical form pairs. They templated and slot-filled physician questions and logical forms on clinical notes and extracted corresponding answers from annotations on clinical notes for tasks like entity extraction and relation learning in the i2b2 challenges (Uzuner et al., 2011).



Question paraphrases

Have this patient's bilirubin changed over time?
What are the recent bilirubin results?
Has the patient had bilirubin testing, if so please give results?
What has this patient's bilirubin been throughout admission?
Does the patient have scanned records for a prior bilirubin?

ABC result
Date: 2116-09-26 04:20:00; Value: 7.3 mg/dL
Date: 2116-09-27 04:30:00; Value: 7.8 mg/dL
Date: 2116-09-28 04:15:00; Value: 10.3 mg/dL
Date: 2116-09-29 04:30:00; Value: 11.4 mg/dL
Date: 2116-09-30 05:27:00; Value: 8.1 mg/dL
Date: 2116-10-01 05:00:00; Value: 6.8 mg/dL
Date: 2116-10-02 04:15:00; Value: 5.9 mg/dL
Date: 2116-10-03 05:00:00; Value: 5.3 mg/dL
Date: 2116-10-04 05:20:00; Value: 5.3 mg/dL
Date: 2116-10-05 04:55:00; Value: 5.1 mg/dL
Date: 2116-10-06 05:30:00; Value: 4.4 mg/dL

Figure 1: Questions (and paraphrases) with answers from MIMIC-III

However, emrQA is restricted to answers within or across clinical notes. Clinical notes are known to capture relations between entities (treatments for problems, side-effects of a drug), signs or symptoms (palpitations), temporal and causal events. On the other hand, structured data in the EHR is considered more reliable for labs results, prescriptions, vitals and other measurements (Hanauer et al., 2015). Hence, a complete EHR QA system should consider data across both these sources in answering a question.

Thus, we propose emrKBQA, a dataset for answering natural language questions from the structured portion of EHR data by mapping questions to logical forms. We demonstrate an instance of using this dataset for question answering using the MIMIC-III KB (a set of question paraphrases and answers from MIMIC shown in Figure 1). The resultant dataset consists of 940,713 question answer pairs from 389 question types (unique instances of questions, i.e., templates) and 52 question/logical

Proceedings of the BioNLP 2021 workshop, pages 64–73
June 11, 2021. ©2021 Association for Computational Linguistics

forms groups (where questions within the group are paraphrases) from 100 patients. We benchmark semantic parsing and answering results on this dataset by learning to map natural language questions to logical forms and retrieving the answer from a KB of patient records. The main contributions of this work are as follows: (1) We develop and release emrKBQA, the first large-scale community-shared dataset for patient-specific QA on structured patient records[1]. (2) emrKBQA will help train models for semantic parsing and answering questions from the structured EHR. This will help us progress towards answering on the EHR as a whole (in conjunction with emrQA). (3) We benchmark state of the art semantic parsing models on the dataset for QA on structured patient records.

2 Related Work

The question answering (QA) problem is usually defined over unstructured texts or structured knowledge bases (KB QA). In case of KB QA, questions are usually mapped to logical forms (or a query language using SQL, SPARQL, etc.) (Zettlemoyer and Collins, 2005; Berant and Liang, 2014) that are then used to retrieve the answer. In the medical domain, there is limited prior work on answering patient-specific questions over structured clinical data.

Roberts and Demner-Fushman (2016, 2015) introduce target logical form definitions and present a rule based method for converting natural language questions over structured data in the EHR into logical forms. They work with a dataset of 446 questions collected during clinician ICU visits and propose an approach using question decomposition, concept recognition and normalization, and rule based semantic parsing. However, the questions and logical forms were not publicly released. In contrast, we present a large-scale community-shared dataset of over 900k generated questions from 52 unique question templates, logical forms and answers.

More recently, Wang et al. (2020) create a new large-scale Question-SQL pair dataset (MIMIC-SQL) on the MIMIC-III dataset, again using the generation process as in Pampari et al. (2018). They propose a deep learning based TRanslate-Edit Model for Question-to-SQL generation that adapts the widely used sequence-to-sequence model to directly generate the SQL query for a given question, and also performs edits using an attentive-copying mechanism. The questions in the dataset are always asked over a patient-cohort such as “how many patients had the diagnosis icd9 code 53190?”. However, the questions in emrQA are specific to a patient. This makes a big difference as the corpus for answering is smaller (limited to the patient’s record, which may include several admissions), the answers may be viewed in conjunction with answers from the unstructured record, the type of questions asked varies, and redundancy and variability in answers to the same question may affect model performance.

Park et al. (2020) construct an EHR QA dataset from MIMIC-III where the question-answer pairs are represented in SQL (table-based) and SPARQL (graph-based). Here again, the questions are defined over patient cohorts; e.g., “What number of married patients suffered from other convulsions?”, making it inherently different from the emrKBQA task. They construct a knowledge graph by relating tables in the database and explore both table-based and graph-based QA (using SPARQL). emrKBQA maps questions to logical forms based on a schema of entities and relations. The tables and columns in the KB are mapped to the entities and attributes in the schema. Logical forms capturing the information need expressed in the question are then instantiated from this schema. Thus, emrKBQA instantiates logical forms from a relational schema (representing entities and relations typically found in the EHR) and facilitates a query language/ resource independent way of representing questions and answering them beyond just individual tables in the KB.

KB-based QA datasets (question semantic parsing) use annotated question and logical form pairs for supervision where the logical forms (that can be then easily be mapped to any query language) are used to retrieve answers from a database (Bordes et al., 2014; Zettlemoyer and Collins, 2005; Berant and Liang, 2014). emrKBQA provides a dataset that can be used to train models to retrieve answers to natural language questions (by mapping them to logical forms) from the structured part of the EHR. The logical forms are instantiated from a schema that captures domain entities, attributes and relations proposed in emrQA (Pampari et al., 2018). We demonstrate the value of the dataset by answering natural language questions posed by physicians

[1] `https://github.com/emrQA/emrKBQA` scripts to generate emrKBQA from MIMIC data.

as follows. We first train state of the art sequence models for semantic parsing to map questions to (query-language agnostic) logical forms. We then map the learned logical forms to the desired query language (SQL) using a deterministic process.

3 Dataset Creation

emrKBQA is generated using a process similar to emrQA. We begin with the same initial question, logical form and template pool as emrQA. However, the question template groups, corresponding logical forms and what constitutes an answer have all been updated by a medical expert to better reflect answering needs.

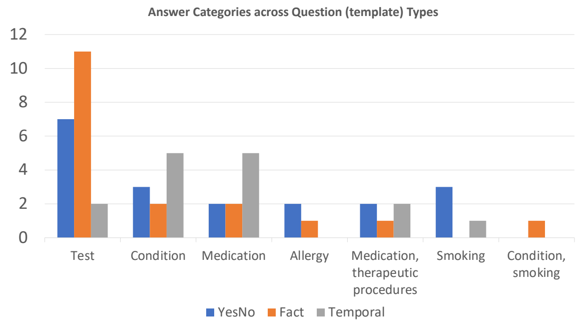


Figure 2: Distribution of answer categories against question template types. Some questions have multiple categories like medication and therapeutic procedure or condition and smoking .

Questions. emrKBQA contains natural language questions posed by physicians at the Veteran's Administration (VA), Mayo Clinic and Cleveland Clinic on patient records (Raghavan et al., 2018). These questions have been transformed into templates by replacing entities with entity-type placeholders (same as emrQA). The dataset consists of 389 such question templates. The placeholders are then slot-filled with appropriate entities from a KB. For instance, "Is the patient on *lisinopril*?" is transformed to: "Is the patient on |*medication*|?" The |*medication*| placeholder is then slot-filled with different medication names from a KB. While the slot-filling is done indiscriminately in emrQA, we constrain the slot-filling by constraining the entity types, wherever possible, with the help of a medical expert. E.g., we filter *Prescriptions* (table) with drug_type (table column) *base* (column value) in slot-filling medication questions. We also filter out certain icd_codes from the diagnoses_icd table in questions with conditions. We process the date field (yyyy-mm-dd, hh:mm:ss) to also insert instances of just month and day, or date without time when slot-filling (along with using the original format). Doing so ensures that the questions are more likely to be naturally asked.

As in this example, the questions are patient-specific and the expected answer is in the structured part of the patient record. Each question template is also assigned one or more question types, which is a new field (not in emrQA) to further categorize question templates in emrKBQA. Question type can take one or more of the following values:

- YesNo = yes/no questions, e.g., "Is |test| value abnormal", "Is the patient on |medication|"
- Temp = temporal or when questions, e.g., "date last |test|"
- Fact = factual or what questions, e.g., "Range of |test|"

A side-effect of the generation process (slot-filling) is that all YesNo questions have a Yes answer. We counter this by also generating questions where the answer will be No. We do this by slot-filling |problem|, |test|, |medication|, |treatment| based on the question and using top 50 most frequently occurring entities in appropriate tables (based on the entity type). Some of these questions are now bound to have No as the answer when applied to our patient set.

The types of questions are a consequence of the questions provided by the physicians who were polled for the initial question set. This was independent of any underlying data and simply based on what they would want to know about their own patients. While several other questions may be answerable on any underlying KB (like MIMIC), we wanted the question set to reflect what an actual physician may want to know from a patient record.

Logical Forms. Logical forms are a structured representation that capture the information need expressed in the question through entities, relations and attributes and are generated as a by-product of the emrQA generation process. They provide a human-comprehensible symbolic representation, linking questions to answers, and help build interpretable models critical to the medical domain (Davis et al., 1977; Vellido et al., 2012). They are formally defined by Pampari et al. (2018) in emrQA. They encapsulate how we are answering a question (since that can be subjective). They are instantiated from a schema representing entities and relations found in the EHR. We use the same

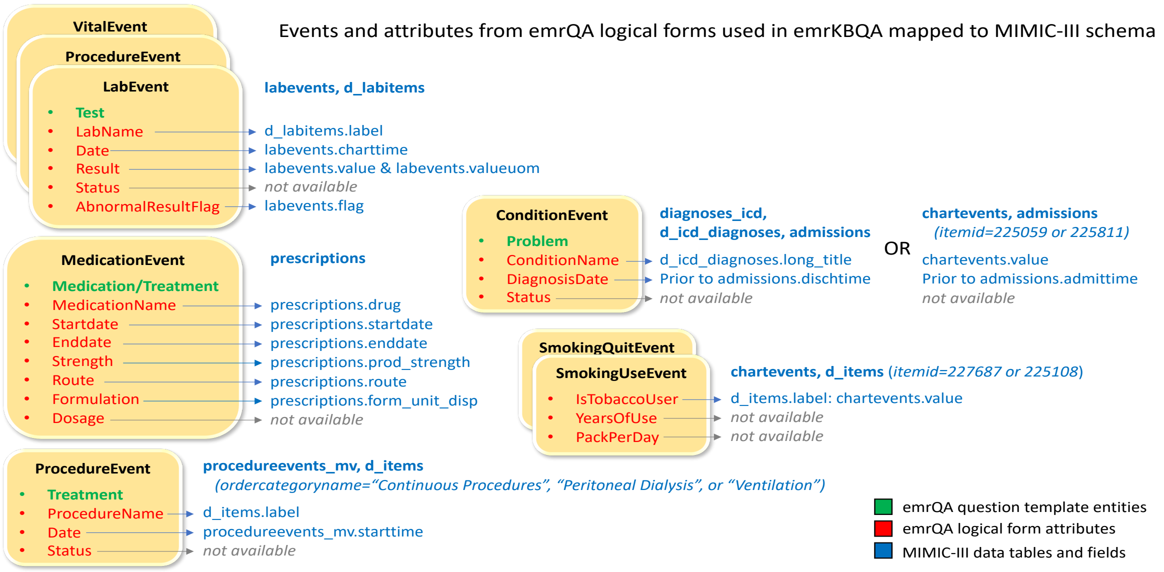


Figure 3: Mapping between emrKBQA schema entities, attributes and tables (yellow boxes) and columns in MIMIC (shown in blue). See MIMIC schema for a description of MIMIC table and column names(Johnson et al., 2016)

schema as Pampari et al. (2018) and map the tables and columns in MIMIC to the schema entities and attributes (see Figure 3).

The schema entities (yellow boxes in Figure 3) represent entities of interest in patient records. In emrQA these are derived from the annotated entities in i2b2 (since emrQA was slot-filled from i2b2 annotations). We use the same entities for emrKBQA as our question set is a subset of emrQA. The structured MIMIC KB does not contain any semantic relations (relates, conducted/reveals, improves, worsens, causes, given/not given (Pampari et al., 2018)). Thus, Figure 3 does not show any of the relations defined in the emrQA schema. An example of the mapping between a schema entity and MIMIC table is as follows. The MedicationEvent (entity that corresponds to Medication and Treatment in our logical form templates) from the schema maps to the Prescriptions table in MIMIC. The entity attributes (shown in red) correspond to the columns in the Prescription table (shown in blue) as illustrated in the figure.

In our example, the logical form for question template "Is the patient on |*medication*|?" would be annotated as "MedicationEvent |medication|", where |medication| would be slot-filled with medication names from the KB. The logical form helps identify appropriate tables, entities and values required from the KB.

Structured data typically factually records lab values, vitals, conditions on admission, and medications but rarely records relations between these entities. In case of emrKBQA, none of the questions that involve resolving relations to answer a question in emrQA are answerable from structured data in MIMIC. However, answering questions about schema entities and attributes requires querying and combining information from multiple related tables in MIMIC.

While logical forms are an outcome of the process used to generate emrQA, they are not essential to answering questions over unstructured data like clinical notes. The more traditional use of logical forms is in answering natural language questions from a structured KB. It is easier to convert a question to logical form than to SQL (which is longer and more complex for most questions, often including multiple nested queries and joins). They provide a query-language agnostic intermediate representation that captures information need expressed in the question using a representation that is perhaps more annotator friendly. Moreover, since logical forms are defined over a schema that captures domain-specific entities and relations, they are independent of the underlying database type or query language.

Question Paraphrase Groups. Question paraphrases are different ways of asking the same thing. The emrKBQA dataset is paraphrase rich with an average of 7.5 paraphrases per question. In emrKBQA, questions that map to the same logical form and share the same question type are considered paraphrases. The dataset has 52 question template groups where each group maps to the same logical form. This is because the answer to a question may vary based on question type even if they map to the same logical form. E.g., Consider the questions in Table 1; the first set of questions are paraphrases since their question type is Fact and they map to the same logical form. So the expected answer is the lab values and date. However, in case of the last question, where the question type is YesNo, the expected answer is a Yes or a No along with the lab values and date. The paraphrases were a natural outcome of the question collection process, where the physicians who were polled phrased the same information need in different ways. Paraphrases may be syntactic variations (word re-ordering) or substitution based (word/ phrase substitution) or a combination of the two.

Paraphrases	Ques Type
Previous |test| levels?	Fact
What is |test| value?	Fact
What is the patient's |test| levels?	Fact
How is his |test| trending?	Fact
Show me a trend of his |test|?	Fact
Has |test| been measured before	YesNo

Table 1: Example question paraphrases that map to the same logical form LabEvent (|test|) [date=x, result=x, sortBy(date)] OR VitalEvent (|test|) [date=x, result=x, sortBy(date)], the first set that also share question type are considered paraphrases.

Answers. Answers in emrKBQA are cell values from a table(s) in the KB. Broadly the answer categories in emrKBQA are Test, Medication, Allergy, Therapeutic Procedures, Conditions and Smoking. Figure 2 shows the distribution of questions across different answer categories. Most questions asking about Test are factual or YesNo whereas Condition and Medication have more questions that are Temporal in nature.

As in emrQA, the answers to questions are derived in a semi-automated manner. Each question is mapped to a logical form that captures the entities and relations that are required to adequately answer the question. This mapping is done by a medical expert. The expert uses an ontology that captures entities, entity attributes and relations in the patient record to define the logical form for a question (we use the same schema as emrQA). The slot-filled logical forms such as, "MedicationEvent|lisinopril|", are mapped to an underlying query language using a deterministic procedure (like SQL) that help retrieve the answer from the KB. The answer to this question would be evidence in the structured data that records the patient taking lisinopril along with some contextual details about the medication. "Yes/No, Start date, End date".

Dataset Generation Process. We use the question/logical form templates from emrQA and filter out templates that cannot be mapped to MIMIC structured data. We then map entity placeholders in the templates to MIMIC columns and populate the placeholders with MIMIC data corresponding to the placeholder entity type. The mapping between entity placeholders and the MIMIC tables and columns[2] is shown in Figure 4. Finally, we extract answers from MIMIC. In the example below, the entity |test| is populated by joining the labevents table with d_labitems (dictionary mapping lab itemids to labels) and retrieving the label field (Hemoglobin), which is used to slot fill the question template and the logical form template. The result for this question is a concatenation of value and valueuom (unit of measurement) from the labevents table; these are sorted by the charttime field. Example questions, logical forms, question type and answer categories are shown in Table 2.

4 Dataset Creation Results

emrKBQA consists of 940,713 question answer pairs over 100 patients, generated from 389 question templates and 52 question type-specific logical form templates[3]. emrKBQA contains an average of 7.5 paraphrases per question type-specific logical form template (ranging from 1 to 55), where a paraphrase is defined as question templates sharing the same question type that map to the same logical form template. Of the generated question answer pairs, 90.9% are test results, 7.8% relate to medications, 1.2% to conditions, and the remaining to other topics (e.g., allergies, tobacco use). The limited size of the medication data can be attributed to the use of emrQA questions as the starting point. emrQA questions are based on an outpatient setting where medication data is available while emrKBQA is from an ICU setting where prescription data is available. Thus several questions about adherence, dosage and frequency of medication were not part of emrKBQA. Only 1% (3,429 rows) of the generated dataset were condition related results since fields such as diagnosis time and relationships

[2] https://mit-lcp.github.io/mimic-schema-spy/

[3] the process can be applied to any number of patients

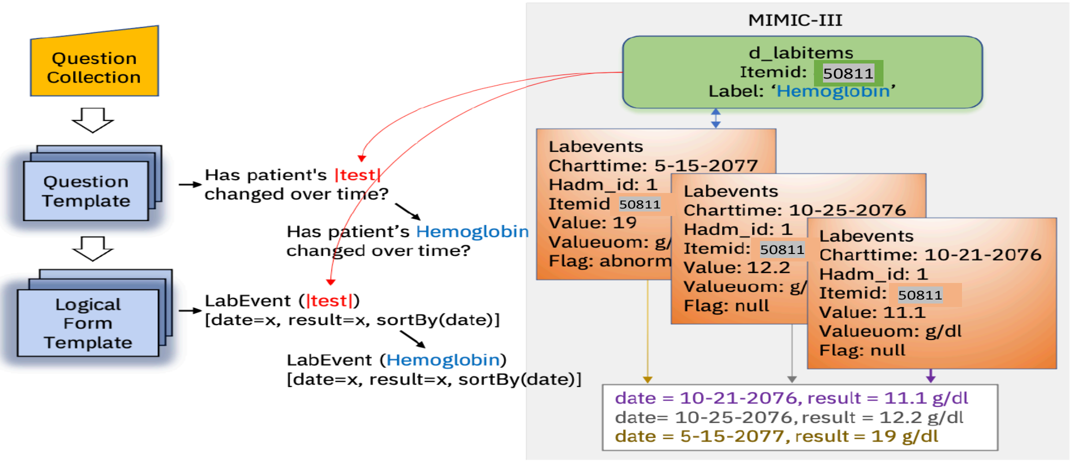


Figure 4: emrKBQA generation process

between treatments and conditions or between medications and conditions are unavailable in MIMIC.

5 Task Definition and Models

Each instance in emrKBQA consists of the follwing elements - question, question paraphrase group, question type, logical form, answer - defined in Section 3. Our goal is to build a model that when presented with a test question on the KB, provides an answer. We achieve this by first modeling the question to logical form learning problem as a semantic parsing task. Here, given an input natural language question, we predict its logical form. Next, we map the predicted logical form to a SQL query in a deterministic manner to retrieve the answer from the KB. The answer is the set of cell values from the underlying KB that answer the question. We detail these two steps in the following sections.

5.1 Semantic Parsing

The task setup for semantic parsing is as follows: given a question in emrKBQA, predict the logical form for that question. As emrKBQA contains several question paraphrases that map to the same logical form, the learning task can be set up in two ways, (1) naive splitting scheme, where input instances are split at random between train and test data, and (2) paraphrase-level splitting scheme, where a question paraphrase seen during train time is not observed in the test set. Thus, the model is tested on whether it can infer the meaning of this question only from its paraphrased forms seen during training. While the paraphrase-level split is more challenging than the naive one, the setting is more realistic. Since the test instances are paraphrases of some training instance, the model is expected to generalize to unseen test instance.

In a previous work, Min et al. (2020) have shown state-of-the-art performance on model generalization for sequence to sequence tasks. They handle unseen sentential paraphrases at test time by incorporating paraphrase detection and generation as auxiliary tasks. In case of paraphrase generation (ParaGen), they sample a question paraphrase during training and learn to generate it along with the main task of logical form prediction. In the paraphrase detection model (ParaDetect), they sample a paraphrase and learn to identify if the sample and the input question are paraphrases by looking at their embeddings in the auxiliiary task. We use the best performing model reported in Min et al. (2020) and perform the following experiments across both splitting schemes: (1) Naive splitting scheme with a baseline model - seq2seq model with copy mechanism (Gu et al., 2016), (2) Paraphrase splitting scheme with a baseline model - seq2seq model with copy mechanism, and (3) Paraphrase splitting scheme with the best-performing ParaGen+ParaDetect model.

5.2 Predicted Logical form to Answer

Finally, the predicted logical form is now mapped to a SQL query to retrieve an answer from the KB. Each question template maps to a logical form template and for each logical form template, we have a corresponding SQL query template. While this mapping is deterministic, the errors in the predicted logical forms require us to use approximate matching functions to map the predicted logical form (template) to the correct logical form template. We

Question	Logical Form	QType	ACat
What were the results of abnormal |test| in |date|?	LabEvent(|test|) [abnormalResultFlag=Y, date=|date|, result=x] OR [{LabEvent(|test|) [date=|date|, abnormalResultFlag=Y]	F	Test
What is the patients |problem| history?	ConditionEvent(|problem|) [diagnosisdate=x] OR SymptomEvent(|problem|) [onsetdate=x]	F	Cond
How long has patient been on |medication|?	MedicationEvent(|medication|) [startdate=x, enddate=x]	T	Med
Has the patient ever been diagnosed or treated for |problem|?	ConditionEvent(|problem|) [diagnosisdate=x] OR [{MedicationEvent(x) OR ProcedureEvent(x)} given ConditionEvent(|problem|)]	YN	Cond

Table 2: Example questions and logical forms across question types Fact(F), Temporal(T), YesNo (YN) and answer categories Test, Condition, Medication

achieve this by matching the by using string similarity measures like edit distance. We then extract the slot filled entity from the predicted logical form and slot fill the SQL query. This query is then run to derive the answer. This answering accuracy is captured in the denotation accuracy metric.

5.3 Experimental Settings

We split emrKBQA dataset according to our two splitting schemes, naive and paraphrase-level, and create two sets of train (70%), dev (10%) and test (20%) datasets. We evaluate the performance of our semantic parsing step using Exact Match (EM) (Min et al., 2020), and our logical form to answer step using Denotation Accuracy (Lin et al., 2019) metrics. EM only considers model outputs that are identical to the labeled ones as correct, while denotation accuracy considers logical forms that return the label answer from the database as correct. We utilize Min et al. (2020)'s public implementation[4] for executing the experiments. We used the default hyperparameters.

5.4 Results

Table 3 presents results of the experiments[5]. The baseline seq2seq with copy model gives high performance in the naive splitting scheme, however the performance drops when we evaluate the model with the paraphrase-level splits. In our experiments, the ParaGen+ParaDetect model provides similar performance to the baseline seq2seq with copy model. This may be attributed to a lack of hyperparameter tuning on out emrKBQA dataset.

Splitting Scheme	Model	EM	Denotation Accuracy
Naive	Seq2seq with copy	0.95	0.96
Paraphrase	Seq2seq with copy	0.83	0.84
Paraphrase	ParaGen + ParaDetect	0.82	0.82

Table 3: Semantic parsing results on paraphrase splits.

For error analysis, we randomly sampled 100 error instances from our best performing seq2seq with copy model predictions. We present the major error categories with examples in Table 4. Almost half of the errors were attributed to questions with multiple entities. In the first example, the two entities "white blood cells" and date "2139-04-01 06:23:00" are merged to "white 06:23:00" in the predicted logical form, leading to an error. Another big chunk of errors can be attributed to incorrect recognition of the entity types present in the question, e.g., whether the entity is of type lab or procedure, or condition or symptom (example 2). To resolve this error, pretraining the model with a named entity recognition objective might be useful. A next set of errors are due to identification of incorrect span of entities (example 3). This error can be attributed to the fact that the the model has not seen the question form in train data (due to paraphrase-level splits). For the remaining error categories, 7% are caused due to attribute errors like min, max, and finally 4% of the errors are

[4] https://github.com/jointparalearning/AdvancingSeq2Seq

[5] Results will vary with different initialization seeds

Question Form	Predicted LF	GT Logical Form	Error category	Perc
what were the results of the abnormal **white blood cells** in **2139-04-01 06:23:00**	labevent (white blood cells) [abnormalresultflag=y, date=2139-04-01 06:23:00, result=x] or procedureevent(white blood cells) [abnormalresultflag=y, date=2139-04-01 06:23:00,result=x] or vitalevent(white blood cells) [date=**white 06:23:00** (result=x)>vital.refhigh] or......	labevent (white blood cells) [abnormalresultflag=y, date=2139-04-01 06:23:00, result=x] or procedureevent(white blood cells) [abnormalresultflag=y, date=2139-04-01 06:23:00,result=x] or vitalevent(white blood cells) [date=**2139-04-01 06:23:00**, (result=x)>vital.refhigh] or	multiple entities	47%
has the patient had a previous **intracerebral hemorrhage**	labevent (intracerebral hemorrhage) [date=x] or procedureevent (intracerebral hemorrhage) [date=x]	conditionevent (intracerebral hemorrhage) [diagnosisdate=x] or symptomevent (intracerebral hemorrhage) [onsetdate=x]	confusion between the entity type	28%
has this patient ever had a documented **chest x-ray** at another va	labevent (documented chest) [date=x] or procedureevent (documented chest) [date=x] or vitalevent (documented chest) [date=x]	labevent (chest x-ray) [date=x] or procedureevent (chest x-ray) [date=x] or vitalevent (chest x-ray) [date=x]	wrong entity span (paraphrase split)	12%
date of **acute bronchitis**	conditionevent (acute bronchitis) [min(diagnosisdate=x)] or symptomevent (acute bronchitis)	conditionevent (acute bronchitis) [diagnosisdate=x] or symptomevent (acute bronchitis) [onsetdate=x]	attribute error	7%
has the patient had a previous **unspecified viral hepatitis c without hepatic coma**	conditionevent (unspecified hepatitis c without hepatic coma) [diagnosisdate=x] or symptomevent (unspecified viral hepatitis c without hepatic coma) [onsetdate=x]]	conditionevent (unspecified hepatitis c without hepatic coma) [diagnosisdate=x] or symptomevent (unspecified viral hepatitis c without hepatic coma) [onsetdate=x]	semantic errors (extra brackets)	4%

Table 4: Error analysis of randomly chosen 100 error instances in the semantic parsing model.

caused due to a long tail of semantic errors like extra brackets, etc.

6 Discussion

Advantages of emrKBQA. emrKBQA is the first large-scale community shared patient-specific QA dataset for answering physician questions from structured patient records. It follows a semi-automated process similar to emrQA (which releases QA pairs on clinical notes), where logical forms are the only expert-provided input. These logical forms lend credibility to the dataset as they capture entities, attributes, and relations required to answer a question and enable slot filling and answer generation. Some highlights of emrKBQA are **(1) Question Quality.** Unlike emrQA, emrKBQA slot-fills entities with discretion by filtering out certain entities based on their attributes (like certain diagnoses based on ICD codes, medications based on drug type). This results in more realistic realization of question instances. **(2) Question Diversity.** The dataset is rich in paraphrases (paraphrase groups have been updated from emrQA) **(3) Dataset Difficulty.** We provide paraphrase-level splits that helps train models that can generalize to unseen paraphrases of the train questions at test time. This is useful in practical settings. As described in the error analysis, in learning to map questions to logical forms, the challenges include recognizing the correct entity spans and types from the question, learning to predict long logical forms, and generating multiple attributes and constraints in the logical form. **(4) Logical forms generated from the same schema as emrQA**, allowing the schema to be a unifying factor across structured and unstructured QA. This allows for future updates in a uniform manner.

Limitations of emrKBQA. (1) Since we wanted the question set to comprise of actual questions asked by physicians, the question set is limited to the initial pool collected from the polled physicians. (2) The dataset is generated in a semi-automated manner that leads to some slot-filled questions that are unlikely to be asked in a real setting. (3) Redundancy of "question form" due to slot filling. Several instances of the same template with different slot-filled entities.

In future versions of the dataset, some of the planned updates include the following: increasing the range of question types, the granularity of questions asked, infuse the need for domain knowledge in understanding a question (using word/ phrase synonyms in slot-filling), better classification of temporal questions based on TimeML, (Pustejovsky et al., 2003), generating more question paraphrases using automated methods (Soni and Roberts, 2019; Min et al., 2020; Neuraz et al., 2018; Dong et al., 2017). While this version of the dataset is generated on randomly sampled 100 patients, we could apply the dataset generation process to any number of patients in MIMIC. It may be interesting to include patient's chosen as per some criteria and contrast answers to similar questions across the chosen cohort.

Differences between emrQA and emrKBQA. emrKBQA is best suited for answering factoid questions such as test results as seen from the results discussed; 87% of emrKBQA (vs 11% of emrQA) comprises test results since test value columns are rarely null. Also, emrKBQA is not limited by annotated clinical notes, which may be a problem if there are very few sources to obtain them. The benefit of emrQA is that it includes questions and answers about medications for problems, response to treatments, temporal constraints and etiology, all of which are unavailable in emrKBQA.

The benefit of a structured dataset such as MIMIC is that explicit values are captured well in tables. Unstructured data may have the answer implicitly stated and may have to be inferred. It also might be incomplete in terms of certain types of crucial information like dates. The limitation of structured data is that it may not capture all types of information. Typically, structured data is unlikely to store symptoms, relations between conditions and symptoms or relations between conditions and treatments. These relations are more likely to be captured by unstructured data.

Question Answering on the entire EHR. emrKBQA is a step in the direction of being able to answer a question anywhere in the EHR, since it utilizes the same schema as emrQA that is used to instantiate logical forms that capture information needs expressed in natural language questions. The answer could now be derived from the structured KB, clinical notes or from both sources in a complementary manner.

7 Conclusion

We create a new large-scale dataset, emrKBQA, for answering patient-specific physician questions from structured patient records. This community-shared release is created in a semi-automated manner and consists of over 900k question-logical form-answer triples, 389 question types (templates), with ≈7.5 paraphrases per question type. We benchmark the dataset and quantify its usefulness in answering questions by training models for semantic parsing of questions to logical forms.

Acknowledgement

This work is supported by MIT-IBM Watson AI Lab, Cambridge, MA USA. We thank the anonymous reviewers for their feedback.

References

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland. Association for Computational Linguistics.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Proceedings of the 2014th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'14, pages 165–180, Berlin, Heidelberg. Springer-Verlag.

Randall Davis, Bruce Buchanan, and Edward Shortliffe. 1977. Production rules as a representation for a knowledge-based consultation program. *Artificial intelligence*, 8(1):15–45.

Dina Demner-Fushman, Wendy Webber Chapman, and Clement J. McDonald. 2009. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42(5):760–772.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

R Scott Evans. 2016. Electronic health records: then, now, and in the future. *Yearbook of medical informatics*, Suppl 1:S48.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

David A Hanauer, Qiaozhu Mei, James Law, Ritu Khanna, and Kai Zheng. 2015. Supporting information retrieval from electronic health records: A report of university of michigan's nine-year experience in developing and using the electronic medical record search engine (emerse). *Journal of biomedical informatics*, 55:290–300.

Ashish K Jha, Timothy G Ferris, Karen Donelan, Catherine DesRoches, Alexandra Shields, Sara Rosenbaum, and David Blumenthal. 2006. How common are electronic health records in the united states? a summary of the evidence: About one-fourth of us physician practices are now using an ehr, according to the results of high-quality surveys. *Health Affairs*, 25(Suppl1):W496–W507.

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.

Kevin Lin, Ben Bogin, Mark Neumann, Jonathan Berant, and Matt Gardner. 2019. Grammar-based neural text-to-sql generation. *arXiv preprint arXiv:1905.13326*.

So Yeon Min, Preethi Raghavan, and Peter Szolovits. 2020. Advancing seq2seq with joint paraphrase learning. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 269–279.

Antoine Neuraz, Leonardo Campillos Llanos, Anita Burgun, and Sophie Rosset. 2018. Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. *arXiv preprint arXiv:1811.09417*.

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, Brussels, Belgium. Association for Computational Linguistics.

Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. 2020. Knowledge graph-based question answering with electronic health records. *arXiv preprint arXiv:2010.09394*.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Preethi Raghavan, Siddharth Patwardhan, Jennifer J Liang, and Murthy V Devarakonda. 2018. Annotating electronic medical records for question answering. *arXiv preprint arXiv:1805.06816*.

Kirk Roberts and Dina Demner-Fushman. 2015. Toward a natural language interface for ehr questions. *AMIA Summits on Translational Science Proceedings*, 2015:157.

Kirk Roberts and Dina Demner-Fushman. 2016. Annotating logical forms for ehr questions. In *LREC... International Conference on Language Resources & Evaluation:[proceedings]. International Conference on Language Resources and Evaluation*, volume 2016, page 3772. NIH Public Access.

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.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. 2012. Making machine learning models interpretable. In *ESANN*, volume 12, pages 163–172. Citeseer.

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

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 658–666, Arlington, Virginia, United States. AUAI Press.