

Contextual Crowd Intelligence

Beng Chin Ooi[†], Kian-Lee Tan[†], Quoc Trung Tran[†], James W. L. Yip[§],
Gang Chen[#], Zheng Jye Ling[§], Thi Nguyen[†], Anthony K. H. Tung[†], Meihui Zhang[†]

[†]National University of Singapore [§]National University Health System [#]Zhejiang University

[†] {ooibc, tankl, tqtrung, thi, atung, zmeihui}@comp.nus.edu.sg

[§] {james_yip, zheng_jye_ling}@nuhs.edu.sg [#]cg@zju.edu.cn

ABSTRACT

Most data analytics applications are industry/domain specific, e.g., predicting patients at high risk of being admitted to intensive care unit in the healthcare sector or predicting malicious SMSs in the telecommunication sector. Existing solutions are based on “best practices”, i.e., the systems’ decisions are *knowledge-driven* and/or *data-driven*. However, there are rules and exceptional cases that can only be precisely formulated and identified by subject-matter experts (SMEs) who have accumulated many years of experience. This paper envisions a more intelligent database management system (DBMS) that captures such knowledge to effectively address the industry/domain specific applications. At the core, the system is a hybrid human-machine database engine where the machine interacts with the SMEs as part of a feedback loop to gather, infer, ascertain and enhance the database knowledge and processing. We discuss the challenges towards building such a system through examples in healthcare predictive analysis – a popular area for big data analytics.

1. INTRODUCTION

Most data analytics applications are industry or domain specific. For example, many prediction tasks in healthcare require prior medical knowledge, such as, identifying patients at high risk of being admitted to the intensive care unit, or predicting the probability of the patients being readmitted into the hospital within 30 days after discharge. Another example from the telecommunication sector is the identification of malicious SMSs requiring inputs from security experts. Building competent tools to effectively address these problems are important, as industrial organizations face increasing pressures to improve outcomes while reducing costs [3].

Existing solutions to industry or domain specific tasks are based on “best practices”. These solutions are *knowledge-driven* (i.e., utilizing general guidelines such existing clinical guidelines or literature from medical journals) and/or *data-driven* (i.e., deriving rules from observational data) [31]. Let us consider the task of identifying the risk factors related to heart failure. The knowledge-driven solution uses risk factors identified from existing clinical knowledge or literature, such as, age, hypertension and diabetes status. However, it may miss out other unknown risk factors specific to the population of interest. The reason is that the guidelines are generic and based on existing knowledge, which results in models that may not adequately represent the underlying complex disease processes in the population with a comprehensive list of risk factors [31]. The

data-driven solution employs machine learning algorithms to derive risk factors solely from observational data. An alternative approach combines the knowledge-driven and data-driven approaches in the data analytics applications [31]. However, there are exceptional situations where it is not easy to capture or formalize, and where neither general guidelines are available nor rules can be derived from data (e.g., in rare conditions). Instead, it is only through many years of experience can subject-matter experts (SMEs) formulate and identify these situations. The challenge then is to be able to capture and utilize such knowledge to effectively support industry/domain specific applications, e.g., improving the accuracy of the prediction tasks.

This paper proposes building the next generation of *intelligent database management systems (DBMSs) that exploit contextual crowd intelligence*. The crowd intelligence here refers to the knowledge and experience of subject-matter experts (SMEs). Although such knowledge is an important component in transforming data into information, it is currently not captured by a structured system. The participants in an intelligent crowd are domain experts rather than “unknown” lay-persons in existing systems that use crowdsourcing as part of database query processing (e.g., CrowdDB [13], Deco [24], Quirk [23], CDAS [12; 22]) and information extraction or knowledge acquisition (e.g., HIGGINS [21] and CASTLE [28]). For applications where data confidentiality and privacy are important (e.g., healthcare analytics), the intelligent crowd may consist of only experts from within the organization, since the tasks cannot be outsourced to external parties. Given that the crowd is known apriori, there is an assurance of user accountability, which translates to an assurance in the quality of the answers. A recent system, called Data Tamer [30], also proposed to leverage on expert crowdsourcing system to enhance machine computation but in the context of data curation. Our proposition differs from Data Tamer in several aspects. First, the target applications of our work (i.e., data analytics) are different from those in Data Tamer (i.e., data curation). Thus, each system needs to address a unique, different set of challenges. Second, the domain experts in our context are also *users/reviewers* of the system. Thus, the experts are likely to take ownership and hence are motivated to improve the accuracy of the analytics and the usability of the applications. This would reduce the need to *localize/customize* the system since the experts/users are continuously interacting with the system; these experts define the “best practices” for the system. For example, doctors in a particular department may use a different convention or notation from another department, e.g., when doctors write “PID” in the orthopedic department, the acronym refers to the “Prolapsed Intervertebral Disc” only and not the “Pelvic Inflammatory Disease”. Clearly, such knowledge can only

be provided by internal domain experts. In contrast, experts in Data Tamer are not the users of the system and hence there is a need to customize/localize the system for different use-cases.

In order to entrench the crowd intelligence into the DBMS, the system needs to keep SMEs as part of the feedback loop. The system can then further utilize feedback provided from the SMEs to infer, ascertain and enhance its processing, thus continuously improving the effectiveness of the system. For example, when predicting the risk of unplanned patient readmissions, the system asks the doctors to label patients who the system has low confidence in predicting their readmissions, and the rules/hypotheses that the doctors used to do the labeling. One example of such an expert rule is that an elderly patient who lives alone and have had several severe diseases is likely to be readmitted into the hospital frequently. The system would then verify or adjust these rules/hypotheses and revert back to the doctors with evidence to support or reject their rules/hypotheses. Such interactions are beneficial to both the system and the doctors. Eventually, the application system evolves over time. SMEs become part of this evolving process by sharing their domain knowledge and rich experience, thereby contributing to the improvement and development of the system. Hence, the experts are more willing and comfortable to use the system to alleviate the burden of their duties.

This work is part of our CIIDAA project on building large scale, Comprehensive IT Infrastructure for Data-intensive Applications and Analysis [2]. Our collaborators are clinicians in the National University Health System (NUHS) [5]. The project aims to harness the power of cloud computing to solve big data problems in the real world, with healthcare predictive analytics being a popular area for big data analytics [26].

Organization. The remainder of this paper is organized as follows. Section 2 presents motivating examples in healthcare predictive analytics. Section 3 discusses the architecture of an intelligent DBMS that aims to embed contextual crowd intelligence. Section 4 elaborates on research problems that we need to address in order to build an intelligent DBMS. Section 5 presents our preliminary results on the problem of predicting the risk of unplanned patient readmissions. Section 6 presents the related work. Finally, Section 7 concludes our work.

2. MOTIVATING EXAMPLES

Let us consider a hospital that has an integrated view of the medical care records of patients as shown in Table 1. The table contains two types of information:

- Structured information, including the case identifier, patient's name, age, gender, race, the number of days that the patient stayed at the hospital during a particular visit (*LengthOfStay*), and the number of days before the patient was readmitted into the hospital after discharge (*Readmission*) ; and
- Unstructured information, i.e., free-text from a doctor's note that contains additional and useful information of a patient healthcare profile such as his past medical history, social factors, previous medications, complaints of patients based on a doctor's investigations, major lab results, issues and progress, etc.

The tuples in this table are extracted from real cases of patients admitted to the National University Hospital (NUH) in Singapore. Healthcare professionals often have queries relating to predicting the severity of patients' condition, such as, identifying patients at

high risk of being admitted to intensive care unit, or predicting the probability of the patients being readmitted into the hospital soon after discharge. There are also queries that monitor real-time data of patients in critical conditions for unusual conditions, such as, whether patients are at high risk of collapsing. With correct predictions, doctors can intervene early to alleviate the deterioration of patient's health outcome. This can potentially reduce the burden of limited healthcare resources in the primary and acute care facilities. For instance, if a patient is at high-risk for unplanned post discharge readmission, he can potentially benefit from close followed-up after discharge, e.g., the hospital sends a case manager or nurse to examine him once every three days. In addition, important queries related to public health surveillance can be answered in a timely fashion. For example, it is critical to provide real-time, early information to alert decision-makers of emerging threats that need to be addressed in a particular population. The ultimate goal of these predictive queries is to predict, pre-empt and prevent for better healthcare outcome.

3. AN INTELLIGENT DBMS FOR BIG DATA ANALYTICS

In this section, we discuss the challenges of addressing big data analytics and present an overview of a hybrid human-machine system for these tasks.

3.1 Challenges of Big Data Analytics

Essentially, many tasks of big data analytics can be viewed as conventional data mining problems, such as, classifying patients into different class labels (high or low risk of being admitted to intensive care units). There are, however, three important aspects that differentiate big data analytics from traditional machine learning problems.

- First, many valuable features for the analytics tasks are stored in unstructured data, for example, doctor's notes [25]. We cannot simply treat these notes as traditional "bag-of-words" documents. Instead, we need powerful tools to extract from these documents the right entities (such as, diseases, medications, laboratory tests) and domain-specific relationships (such as, the relationship between a disease and a laboratory test). The text in unstructured data has to be contextualized to each organization's practice, e.g., doctors in a particular department may use a different convention or notation from another department.
- Second, there is usually a lack of training samples with well-defined class labels. For instance, when predicting the risk of committing suicide for each patient, the total number patients known to have committed suicide (i.e., class 1) is very small. However, it does not mean that all the remaining patients did not commit suicide (i.e., class 0). Hence we need to infer the correct class labels for these patients. This problem also occurs in other domains such as home security and banking. For example, one important task that many national security agencies need to perform is identifying persons or groups of people who will likely commit a crime [4]. In this setting, the agency maintains a very small set of people who have committed crime. However, we cannot simply assume that the remaining people are not likely to commit crime. As before, we need to infer the correct class labels for these people. Another example is in telecommunication, where a service provider wants to predict whether an SMS is malicious. In this case, we do not

CaseID	Name	Age	Gender	Race	LengthOfStay	Readmission	Doctor's note
Case 1	Patient 1	71	Female	Chinese	5	20	PMH: 1 IHD - on GTN 0.5mg prn 2 DM - on Metformin 750mg - HbA1c 7.5% 09/12 3 HL Stays with son ...
Case 2	Patient 2	60	Male	Malaysian	10	20	Social issues: Single, no child Used to live with friend in a shophouse Now at sheltered home since Sept 2011. No next-of-skin or visitor. ...

Table 1: Medical care table

have any predefined class labels and might need to ask security experts to provide the class labels for some sample cases.

- Lastly, data in different domains (e.g., healthcare, telecommunication, home security) is expected to grow dramatically in the years ahead [26]. For instance, patients in intensive care units are constantly being monitored, and their historical records have to be retained. This can easily result in hundreds of millions of (historical) records of patients. As another example, during a mass casualty disaster (e.g., SARS, H5N1), there is an overwhelming number of patients who have to be monitored and tracked, and information about each patient is huge by itself. Furthermore, streaming data arrive continuously, e.g., new data from the real-time data feed are constantly being inserted. Hence, the system in healthcare setting must provide the real-time predictions, e.g., predicting the survival of patients in the next 6 hours.

The three above mentioned aspects call for a new generation of intelligent DBMSs that can provide effective solutions for big data analytics. Our proposition of exploiting contextual crowd intelligence is, we believe, a big step towards this goal.

3.2 Contextual Data Management

The central theme of crowd intelligence is to get domain experts engaged as both the participants to fine tune the system and the end-users of the system. Figure 1 presents an intelligent system that exploits contextual crowd intelligence for big data analytics. The system first builds a knowledge base that will be subsequently used for the analytics tasks based on historical data, domain knowledge from SMEs (e.g., doctors), and other sources such as general clinical guidelines. Each source contributes to build some “weak classifiers”. The system needs to combine these classifiers to derive a final classifier that achieves a high level of accuracy for prediction purposes. The system also needs to go through several iterations of interaction with the experts to refine, for example, the final classifier. As such, the experts participate in the entire process in fine tuning the system and decide on the “best practices”. When real-time data or feed arrives, the system performs the prediction on-the-fly and alerts the experts immediately. Hence, the experts become the end-users of the system.

We have developed the epiC system [1; 10; 19] to support large scale data processing, and are extending it to support healthcare analytics. Figure 2 shows the software stack of epiC. At the bottom, the storage layer supports different storage systems (e.g.,

Hadoop Distributed File System (HDFS) and a key-value storage system, ES² [8]) for both unstructured and structured data. The next layer (which is the security layer) enables users to protect data privacy by encryption. The third layer (which is the distributed processing layer) provides a distributed processing infrastructure called E³ [9] that supports different parallel processing logics such as MapReduce [11], Directed Acyclic Graph (DAG) and SQL. The top layer (which is the analytics layer) exploits the contextual crowd intelligence for big data analytics. The details of this layer are shown in Figure 1. In Figure 2, KB is the knowledge base and iCrowd is the component that interacts with the domain experts. Different components of the analytics layer (e.g., scalable machine learning algorithms) can process their data with the most appropriate data processing model and their computations will be automatically executed in parallel by the lower layers.

In the remaining of this paper, we focus only on the analytics layer. For more details of the other layers of the epiC system, please refer to [1; 10; 19].

4. RESEARCH PROBLEMS

In this section, we elaborate on the research problems that we need to address in order to build an intelligent system for big data analytics.

4.1 Asking Experts The Right Questions

Given a large volume of data and a limited amount of time that domain experts can participate in building the systems, we need to ask the experts the right questions. In the context of healthcare analytics, we plan to ask the following domain knowledge from doctors.

- **Labelings.** The system asks doctors to label tuples that the system has low confidence in performing the prediction task. There are two important issues here. First, doctors have different levels of confidence when answering different questions, i.e., doctors are reluctant to assess patient profiles that they do not have specialties. Second, since there is so much information about patients, selecting the relevant feature of each patient to present to the doctors in order not to overwhelm them is also a major issue.

In essence, what we need is a diverse set of labeled patients that covers the whole data space as much as possible. One possible solution is to group similar patient profiles together and show these groups to doctors. The purpose is to let the doctors select the groups of patients that they are comfortable in providing the labels. In addition, for each

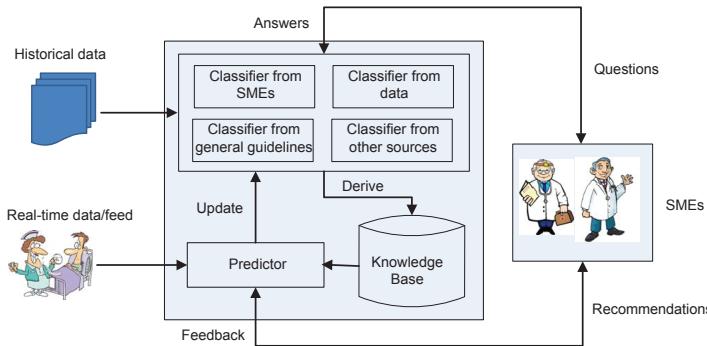


Figure 1: Contextual crowd intelligence for big data analytics.

group, we present only the features which the patients in the group have similar values. In this way, we can avoid overwhelming the doctors with information. Note that, in some cases, we need to perform hierarchical clustering to reduce the number of patients shown to the doctors each time. Selecting the right clustering algorithms and developing effective visualization tools to present patient's profiles are important here.

- **Rules/Hypotheses.** The system collects expert rules/hypotheses that the doctors used to do the labeling. For example, to predict the risk of unplanned patient readmissions, the doctors suggested a hypothesis that social factors and the status of the diseases are important risk indicators for readmission. The system would then verify or adjust these hypotheses and revert back to the doctors with evidence to support or reject their hypotheses. Such interactions are beneficial to both the system and the doctors.
- **Inferred implicit knowledge.** The system can also infer implicit and valuable knowledge based on the answers/reactions of the domain experts. For instance, if the doctors label two patients who belong to a given cluster differently, then the system can adjust the distance function used to compute the similarity between two patients, and thus infer which features are more important. Such knowledge is implicit as the doctors themselves may not be aware of.

We can also ask the same kind of questions for the analytics tasks in other domains. For instance, to predict malicious SMSs, we need to select a small set of messages (by utilizing some clustering algorithms) and ask the experts to provide labels for these samples. We also collect rules and heuristics that the experts utilize to label the SMSs.

4.2 Extracting Domain Entities From Unstructured Data

Feature selection is very important for any machine learning task and can greatly affect the algorithm's quality. Processing doctor's notes for extracting important features is an inevitably important step for healthcare analytics problems. There are several state-of-the-art Natural Language Processing (NLP) engines for processing clinical documents, such as, MedLEE [14] and cTAKES [27]. These engines process clinical notes, identifying types of clinical entities (e.g., medications, diseases, procedures,

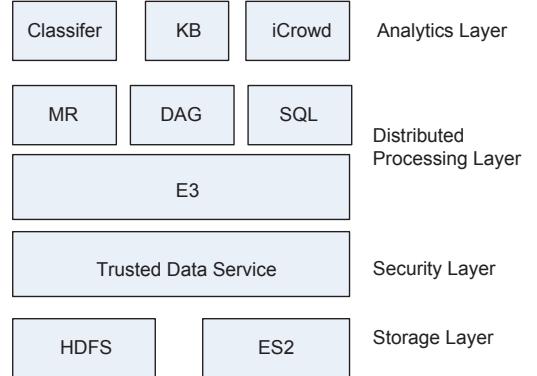


Figure 2: The software stack of epiC for big data analytics.

lab tests) from various medical dictionaries (a.k.a. knowledge base), such as, the Unified Medical Language System (UMLS) [6]. We now discuss several problems raised due to the nature of the unstructured data and the incompleteness of the knowledge base, and subsequently discuss a hybrid human-machine approach to solve these problems. The discussion uses the following running example. We run cTAKES on the doctor's note of patient 1 (in Table 1), and obtain the following clinical entities: (1) *diseases*: IHD (Ischemic Heart Disease) and DM; (2) *medications*: GTN and Metformin; and (3) *laboratory test*: HbA1c.

Ambiguous mentions. In many cases, a mention in the free text may refer to different domain entities. For instance, in the running example, “DM” refers to two different diseases “Dystrophy Myotonic” and “Diabetes Mellitus”. We note that this problem is not uncommon as doctors tend to use abbreviations in their notes. For example, “CCF” refers to either “Congestive heart failure” or “Carotid-Cavernous Fistula” diseases; “PID” refers to either “Prolapsed Intervertebral Disc” or “Pelvic Inflammatory Disease”. There are also cases where only human but not the machine can understand the meaning of some mentions in the text. For example, assuming that we are extracting the social factor of patients in Table 1. It is rather easy to extract the social factor for patient 1, since the text contains the phrase “stays with son”. However, it is challenging, if not possible, for the machine to extract the social factor for patient 2. The reason is that the paragraph contains several different keywords relating to the social factor such as “single”, “no child”, “live with friend”, “sheltered home”, “next-of-kin”.

Incomplete knowledge base. The knowledge base is incomplete for the following reasons. First, the terms used in the doctor's notes could be specific within a country or a particular hospital, whereas the existing knowledge bases may only cover the universal ones. Thus, these terms do not exist in the dictionary. One example is the term “HL” in our running example, which refers to the “Hyperlipidemia” disease but is not captured in UMLS. Second, the relationships between entities covered in existing medical knowledge bases (like UMLS) are far from complete. In the running example, the fact that the medication Metformin is used to treat Diabetes Mellitus (DM) is also missing in UMLS. The relationships that exist between domain entities can be used to derive implicit and useful information. For instance, from the laboratory result of the lab test HbA1c, we can infer whether the DM condition is well-controlled (i.e., the relationship between a disease and a lab test).

A hybrid human-machine approach. To infer the correct entities from unstructured data, a hybrid human-machine solution should be employed. The system can leverage the information from the knowledge base (e.g., UMLS) together with the implicit information (signals) inherent in the unstructured data (e.g., doctor’s notes) to improve the accuracy of its inference process and enhance the knowledge base as well. The system will pose questions to the healthcare professionals for verification. Based on the answers from the experts, the system adjusts its inference results. The inference process gets more accurate and complete as the system runs more iterations. Meanwhile, the knowledge base becomes more comprehensive and customized to each organization’s practice. More specifically, in our running example:

- Since “DM” is attached with the laboratory test “HbA1c” in the paragraph, the machine conjectures that “DM” would refer to the “Diabetes Mellitus” disease only. The reason is that HbA1c is a laboratory test that monitors the control of diabetes and HbA1c does not have any relationship with the other disease related to “DM” (i.e., “Dystrophy Myotonic”).
- To correctly infer the disease “Hyperlipidemia” for “HL”, the machine infers a pattern of “*num d*” where *num* is a fraction annotation and *d* is a disease. (“1 IHD” and “2 DM” are two examples.) The machine then infers that “HL” may refer to a disease since the phrase “3 HL” follows the pattern. The machine then poses a question to a doctor: which disease “HL” represents for? In this case, the doctor confirms that “HL” represents for the “Hyperlipidemia” disease. Based on the answer, the machine adds the mapping between the mention “HL” and the disease “Hyperlipidemia” to the knowledge base. Hence, the knowledge base becomes more comprehensive and customized to NUH’s practice.
- To identify the missing relationship between the medication Metformin and the disease DM, the machine infers a pattern of “*d on med*”, where *d* is a disease, *med* is a medication and *med* is used to treat *d*. (“IHD on GTN” is an example.) The machine conjectures that there should have a relationship between DM and Metformin, since the phrase “DM on Metformin” follows the pattern. The machine then verifies this inference with the doctors. The doctors confirm that they typically write the medications that are used to treat a disease right next to the disease, and connect these relationships by the preposition “on”. Clearly, such rule is very useful – the machine will then infer other missing relationships using this expert rule with fewer questions being posed to the doctors.
- To derive the social factor for patient 2, the machine can first attempt to derive the information using a simple strategy such as analyzing the NLP structure of sentences containing patterns like “stay with”, “live with”. For complicated cases when the machine cannot find out the information, we need to tap on the knowledge of the experts.

4.3 Combining Multiple Weak Classifiers

We can obtain different classifiers from multiple sources such as classifiers built based on the observational data, rules used by the doctors and general clinical guidelines. Each source of knowledge can be considered as a “weak classifier” and the task is to combine these classifiers to derive a final classifier that achieves a very high accuracy in prediction.

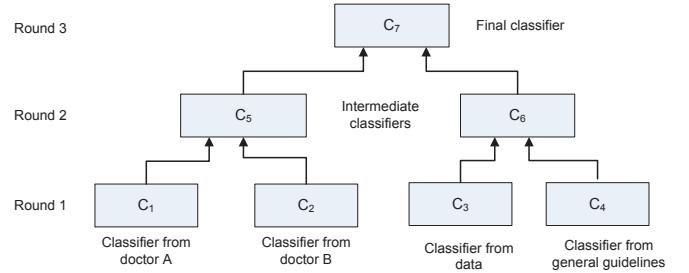


Figure 3: An example of several rounds of learning for healthcare predictive analytics

There are many ways to achieve the goal. Figure 3 shows an example of a process consisting of three rounds of learning for the task of predicting the severity of patients. In the first round, the system computes four classifiers: C_1 and C_2 are the classifiers derived from rules provided by SMEs (i.e., doctors); C_3 is the classifier derived from historical data; and C_4 is the classifier derived from clinical guidelines. It is essential to resolve disagreeing opinions from various sources. There are several ways to combine different classifiers, such as, using majority-voting for the outputs of different rules/classifiers or combining features being used in different input classifiers.

It is likely that all the classifiers built after the first round do not agree with each other for the prediction tasks. Thus, in this example, the system performs two additional rounds of learning to improve the accuracy of the classifier. It is also possible that there is no way to reconcile the classifiers, i.e., there will be multiple different classifiers. In such situations, it may be necessary to “rank” the results of the different classifiers, and pick the answer that is ranked highest. How to do this is an open question.

4.4 Scalable Processing

Big data analytics is characterized by the so-called 3V features: Volume - a huge amount of data, Velocity - a high data ingestion rate, and Variety - a mixed of structured, semi-structured and unstructured data. These requirements force us to rethink the whole software stack to address big data analytics efficiently and effectively, ranging from the storage layer that should manipulate both structured and unstructured data to application layer that should support scalable machine learning algorithms. To illustrate the points, let us reconsider the problem of predicting the malicious SMSs. The collection of SMSs is huge, e.g., in the order of hundreds of tera-bytes. As discussed in Section 4.1, we need to pick a set of SMSs for domain experts to label. Conventional clustering algorithms may not work well here as we need to handle such a large amount of data. The problem is even more challenging in our context, as we need to frequently get the domain experts involved in building the system. The delay from human beings’ reaction may be a large factor affecting the low latency of the system.

The scalability of the problems is also in terms of high-dimensional data space. Our data set inherently contains a large number of features. For instance, there are different information about patients such as thousands of different diseases and lab tests. One solution to reduce the dimensions is to group these attributes *semantically*, e.g., grouping together different diseases that share a same “root”. For instance, the Hypertension disease, Hypotension disease and Ischaemic Heart disease can be grouped together under the category of Cardiovascular disease.

Clearly, to perform such tasks, we need to consult the domain experts as different hospitals/doctors may have different opinions/reasoning in performing this task. This is, again, an example of getting the domain experts involved in building the systems.

4.5 Engaging Expert Users

As the system needs to interact with SMEs frequently, it is important to engage the experts along the process of building and using the system. The system should provide several functionalities for this purpose:

- A user-friendly interface for the experts to provide their inputs such as rules, hypothesis, labels, etc.
- The system should provide not only the final outcome (e.g., whether the patient is at high/low risk of being sent to ICU) but also the reasons that drive its decision. Therefore, keeping track of the provenance of the knowledge is important. For instance, when the system makes a decision that differs from experts' opinions, the system should be able to trace back whether the mismatch is mainly due to the use of some general guidelines, or due to other experts' opinions.
- Presenting feedback to the experts. For instance, the system can explain how well an expert performs compared to other colleagues. As another example, the system can reveal comments and annotations by other experts to see whether an expert would change her decision. It is also interesting to present new patterns of knowledge that an expert may lack and potentially educate her.

5. PRELIMINARY RESULTS

We are studying the problem of predicting the probability of patients being readmitted into the hospital within 30 days after discharge. We refer to the task as *readmission prediction* for short. We use the clinical data drawn from the National University Hospital's Computerized Clinical Data Repository (CCDR) and focus only on the *elderly patients* (i.e., patients with age older than 60) admitted to the hospital in 2012. The table used for the prediction task is the medical care table¹ that has similar schema as the one presented in Table 1. There are in total 29049 elderly patients admitted to NUH in 2012, where 5658 patients readmitted within 30 days, i.e., the proportion of patients who were readmitted (i.e. class label 1) is 0.188.

5.1 Interacting with Domain Experts

We have been getting the doctors involved in the following tasks.

Hypothesis/Rules. Our clinician collaborators have suggested a hypothesis that the following features (indicators) might be important for the readmission prediction:

- Social-economic factors, e.g., who are the care-givers and the patient's economic status.
- Lab findings. We should extract the lab findings that the doctors mentioned in their notes instead of using the labs recorded in the structured data in CCDR. The reason is that patients typically have hundreds of lab tests but only a small

¹To derive the medical care table, we joined information from various relations in CCDR, including: Discharge Summary, Patient Demographics, Visit and Encounter, Lab Results and Emergency Department.

	# actual class 1	# actual class 0
#predicted class 1	1071	1321
#predicted class 0	4587	22070

(a) Using only structured features

	# actual class 1	# actual class 0
#predicted class 1	2679	4250
#predicted class 0	2979	19141

(b) Using both structured and derived features

Table 2: The accuracy of our classifier.

number of them is important and is captured in the doctor's notes. As a result, selecting lab findings mentioned by doctors naturally reduces the dimensions of the data set.

- Comorbidity influence, i.e., we should take into account the past medical history of the patient together with the disease status (whether the disease has been well-controlled).

Participants in a crowd-sourcing system. We adopted a hybrid human-machine approach to extract the social factors and lab findings from doctor's free-text notes.

To extract the social factors, we use an NLP technique to analyze sentences containing phrases related to the social factor such as "live (with)", "stay (with)", "main care-giver" to pinpoint some keywords such as "daughter", "family", "spouse", etc. The system then asks the doctors to handpick a set of predefined categories of social factors. For instance, living with family and taking care by professional helpers (e.g., maid, domestic helpers) are in a same group. As another example, living alone and living in a community nursing home are in a same group. The system also performs a postprocessing step to pull out cases that can be assigned more than one category of social factors. The system then asks the doctors to label these cases manually. (There are about 200 cases that need to be manually labeled.)

To extract the lab findings, the system first uses a simple pattern matching technique to extract all possible lab tests mentioned in the note. For instance, if the note contains a pattern of the form "*word num*" where *word* is some word and *num* is a number, then *word* is a candidate lab test. A *word* is a correct lab test if it exists in the medical dictionary with the category of lab tests. For the "false" lab tests that are currently not present in the dictionary and appear frequently in the notes, the system asks the doctors to verify them. As a result, there are some actual lab tests that are missing in the dictionary such as "TW", which is a local convention used inside NUH.

Extracting medical concepts. We run the cTAKES NLP engine over the UMLS dictionary to extract the past medical history of a patient. We are in the process of developing algorithms to improve the accuracy of extraction (to resolve problems mentioned in Section 4.2). Thus, we use the number of diseases that the patient has as an indicator instead of the actual diseases.

5.2 Results

After interacting with the doctors to extract relevant features, we obtained two sets of features for the prediction task:

- *Structured features:* patients' demographics (age, gender, race), the number of days that the patient stayed at the hospital, the number of previous hospitalizations, and the

number of prior emergency visits in the last six month before admission.

- *Derived features* from free-texts (We refer to these features as derived features for short): social factors, lab findings, and past medical history (i.e., diseases).

We used WEKA [15] to run a 10-fold cross-validation and the Bayesian Network classifier to construct a readmission classifier². Table 2 reports the accuracy of the prediction across all the 10 validation data. If only structured features are used to build the classifier (Table 2(a)), the resulting classifier can correctly predict 1071 cases that are readmitted (within 30 days). The precision and recall in this case are 0.448 and 0.189, respectively. Meanwhile, if both structured and derived features are used to build the classifier (Table 2(b)), the resulting classifier can correctly predict 2679 cases that are readmitted. The precision and recall are 0.387 and 0.473 respectively. Clearly, the recall has been improved significantly with the usage of the derived features from the free-text doctor’s notes. The result is also very promising when we compared it to the result handled manually by domain experts such as physicians, case managers, and nurses [7]. The recall reported in [7] is in the range [0.149, 0.306]. The conclusion in [7] is that care-providers were not able to accurately predict which patients were at highest risk of readmission. However, we believe that a hybrid machine-human solution would greatly alleviate the problem.

We would like to emphasize that there are many rooms to further improve the accuracy of the prediction such as enhancing the feature extraction process, employing additional features, such as, disease status, specific diagnoses, medications, and using special classifiers for highly-imbalanced data set.

6. RELATED WORK

Related works to our proposition can be broadly classified into the following three categories.

Existing solutions for industry/domain specific applications. Existing solutions are currently built based on “best practices”. One direction is knowledge-driven approach that is based on general guidelines such as clinical guidelines, e.g., IBM Watson [3]. Another direction is data-driven approach that is based on “rules” extracted from the observational data, e.g., [16; 18; 20]. Recently, IBM proposes to combine the strengths of the two directions [31]. However, these solutions have not explored the exceptionally complicated rules/patterns that can only be provided by *internal domain experts* with years of working experience. Our research aims to fill this gap: we seek to engage the experts as users of the system, and tap on their expertise to enhance the database knowledge and processing. There are several benefits of employing internal domain experts. First, we do not need to customize/localize the system for different use-cases; they themselves define the “best practices” for the system. Second, in terms of the data used to build the knowledge base, our system mainly bases on observational data and knowledge provided by domain experts; whereas others (e.g., IBM Watson) need to process a much larger amount of inputs such as medical journals, white papers, medical policies and practices, information in the web, etc. Third, the system should become more “intelligent” over times when the expert users continuously enhance the system with their expert knowledge.

²We also used other classifiers such as decision tree, rule-based classifier, SVM, etc and observe that the Bayesian Network classifier provides the best result.

Crowdsourcing in database. There has been a lot of recent interest in the database community in using crowdsourcing as part of database query processing (e.g., CrowdDB [13], Deco [24], Quirk [23], CDAS [12; 22]). As discussed, the intelligent crowds in our context are domain experts (rather than lay-persons in the existing crowds) who are also users/reviewers of the system. Furthermore, exploiting intelligent crowd can be much more collaborative in nature. In typical crowdsourcing, the crowds are not aware of each other’s answers. But in our context, we can actually go through several iterations and see whether the experts will change their decisions when they are provided with comments and annotations by other experts.

A recent system, called Data Tamer [30], also leveraged expert crowdsourcing system to enhance machine computation but in the context of data curation. As discussed in Section 1, the key difference between our proposition and Data Tamer lies in the fact that the domain experts in our context are also *users/reviewers* of the system. Thus, the experts are likely to take ownership and hence are motivated to improve the accuracy of the analytics and the usability of the applications. This would reduce the need to localize/customize the system. Also, each system needs to address a different set of challenges, since the targeted applications are different.

Active learning. In the active learning model, the data come unlabeled but the goal is to ultimately learn a classifier (e.g., [17; 29; 32]). The idea is to query the labels of just a few points that are especially informative in order to obtain an accurate classifier. The labels are obtained from highly-trained experts (e.g., doctors). The scope of our proposition is much more general than active learning in the following points. First, we would like to exploit as much domain knowledge from experts as possible, not restricting to only the class labels as in active learning. For instance, rules and hypotheses provided by experts with many years of experience must be exploited in several cases. Second, active learning focuses on getting a better classifier so the query points presented to the crowd are usually those data points that are at the boundary of the separating plane. However, these are also the data points that the experts are usually not very clear about. As such, we need to be able to identify additional information that should be provided for the experts to be able to make an informed decision. Lastly, we need to handle a large amount of data whereas existing solutions on active learning usually deal with small data set.

7. CONCLUSION

Each of us is a subject-matter expert (SME) of our profession, and we carry with us a vast amount of knowledge and insights not captured by a structured system. This might have explained the emergence of Knowledge Management systems. However, there are many rules and exceptional cases that can only be formulated by experts with many years of experience. Such rules, when properly coded, can help in facilitating contextual decision making. This paper envisions a more intelligent DBMS that captures such information or knowledge. At the core, the system is a hybrid human-machine database processing engine where the machine keeps the SMEs as part of the feedback loop to gather, infer, ascertain and enhance the database knowledge and processing. This paper discussed many open challenges that we need to tackle in order to build such a system.

8. ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation, Prime Minister’s Office, Singapore under Grant No.

NRF-CRP8-2011-08. We thank Associate Professor Gerald C.H. Koh and Dr. Chuen Seng Tan (Saw Swee Hock School of Public Health, National University Health System) for sharing with us domain knowledge in healthcare.

9. REFERENCES

- [1] <http://www.comp.nus.edu.sg/~epic>.
- [2] The comprehensive it infrastructure for data-intensive applications and analysis project. <http://www.comp.nus.edu.sg/~ciidaa/>.
- [3] Ibm big data for healthcare. <http://www.ibm.com>.
- [4] The minority report: Chicago's new police computer predicts crimes, but is it racist? <http://www.theverge.com/2014/2/19/5419854/the-minority-report-this-computer-predicts-crime-but-is-it-racist>.
- [5] National university health system. <http://www.nuhs.edu.sg/>.
- [6] Unified medical language system. <http://www.nlm.nih.gov/research/umls/>.
- [7] N. Allaudeen, J. L. Schnipper, E. J. Orav, R. M. Wachter, and A. R. Vidyarthi. Inability of providers to predict unplanned readmissions. *J Gen Intern Med*, 26(7):771776.
- [8] Y. Cao, C. Chen, F. Guo, D. Jiang, Y. Lin, B. C. Ooi, H. T. Vo, S. Wu, and Q. Xu. Es2: A cloud data storage system for supporting both oltp and olap. In *ICDE*, pages 291–302, 2011.
- [9] G. Chen, K. Chen, D. Jiang, B. C. Ooi, L. Shi, H. T. Vo, and S. Wu. E3: an elastic execution engine for scalable data processing. *JIP*, 20(1):65–76, 2012.
- [10] G. Chen, H. Jagadish, D. Jiang, D. Maier, B. Ooi, K. Tan, and W. Tan. Federation in cloud data management: Challenges and opportunities. *TDKE*, 2014.
- [11] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1), Jan. 2008.
- [12] J. Fan, M. Lu, B. C. Ooi, W.-C. Tan, and M. Zhang. A hybrid machine-crowdsourcing system for matching web tables. In *ICDE*, pages 976–987, 2014.
- [13] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *SIGMOD Conference*, pages 61–72, 2011.
- [14] C. Friedman, P. O. Alderson, J. H. Austin, J. J. Cimino, and S. B. Johnson. A general natural-language text processor for clinical radiology. *JAMIA*, 1(2):161–174, 1994.
- [15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.
- [16] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [17] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, pages 417–424, 2006.
- [18] A. Hosseinzadeh, M. T. Izadi, A. Verma, D. Precup, and D. L. Buckeridge. Assessing the predictability of hospital readmission using machine learning. In *IAAI*, 2013.
- [19] D. Jiang, G. Chen, B. C. Ooi, K.-L. Tan, and S. Wu. epic: an extensible and scalable system for processing big data. In *PVLDB*, 2014.
- [20] P. S. Keenan, S.-L. T. Normand, Z. Lin, E. E. Drye, K. R. Bhat, J. S. Ross, J. D. Schuur, B. D. Stauffer, S. M. Bernheim, A. J. Epstein, Y. Wang, J. Herrin, J. Chen, J. J. Federer, J. A. Mattera, Y. Wang, and H. M. Krumholz. An administrative claims measure suitable for profiling hospital performance on the basis of 30-day all-cause readmission rates among patients with heart failure. *Circ Cardiovasc Qual Outcomes*, 1(1):29–37, 2008.
- [21] K. S. Kumar, P. Triantafillou, and G. Weikum. Human computing games for knowledge acquisition. In *CIKM*, pages 2513–2516, 2013.
- [22] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: a crowdsourcing data analytics system. *PVLDB*, 5(10):1040–1051, 2012.
- [23] A. Marcus, E. Wu, S. Madden, and R. C. Miller. Crowd-sourced databases: Query processing with people. In *CIDR*, pages 211–214, 2011.
- [24] A. G. Parameswaran, H. Park, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: declarative crowdsourcing. In *CIKM*, pages 1203–1212, 2012.
- [25] S. Perera, A. Sheth, K. Thirunarayan, S. Nair, and N. Shah. Challenges in understanding clinical notes: Why nlp engines fall short and where background knowledge can help. In *CIKM Workshop*, 2013.
- [26] W. Raghupathi and V. Raghupathi. Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, 2014.
- [27] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. K. Schuler, and C. G. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *JAMIA*, 17(5):507–513, 2010.
- [28] T. K. Sean Goldberg, Daisy Zhe Wang. Castle: Crowd-assisted system for textual labeling & extraction. *HCOM*, 2013.
- [29] B. Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2010.
- [30] M. Stonebraker, D. Bruckner, I. Ilyas, G. Beskales, M. Cherniack, S. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The data tamer system. In *CIDR*, 2013.
- [31] J. Sun, J. Hu, D. Luo, M. Markatou, F. Wang, S. Edabolahi, S. E. Steinhubl, Z. Daar, and W. F. Stewart. Combining knowledge and data driven insights for identifying risk factors using electronic health records. In *AMIA*, 2012.
- [32] J. Wiens and J. Guttag. Active learning applied to patient-adaptive heartbeat classification. In *NIPS*, pages 2442–2450, 2010.