Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 12 Issue 12 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Data Mining in Clinical Practices Guidelines

By Mayura Kinikar, Harish Chawria, Pradeep Chauhan & Abhijeet Nashte
Maharashtra Academy of Engineering Alandi, Pune

Abstract - This paper proposes text mining of clinical practices to extract decision-making steps. These steps should be formed in- logical functions capable of branching on different plan set on some deciding variables. The probable action sequence will be notified on the data of patient given to the conditions of clinical guideline and this will also give critical conditions that need immediate attention. In this project medical grammar rules are applied to extract key decision making steps from the clinical guidelines. In the first step lexical analysis is performed to key- words like 'if this then perform this, all the medical terms will be identified and this extracted rule set will be used to create a XSLT file. The patient data in form of an XML file will be then applied to the XSLT transformations or rule sets to derive final result of action plan specific to that patient.

Keywords : *Clinical Data Repository(CDR), Virtual Medical Record(VMR), Abstract Syntax Notation(ASN), Electronic Medical Records(EMR), Medical Logic Modules(MLM), HealthCare Data Dictionary(HDD).*

GJCST-C Classification: *H.2.8*

Strictly as per the compliance and regulations of:

Data Mining in Clinical Practices Guidelines

Mayura Kinikar [α], Harish Chawria [α], Pradeep Chauhan [α] & Abhijeet Nashte [α]

Abstract - This paper proposes text mining of clinical practices to extract decision-making steps. These steps should be formed in- logical functions capable of branching on different plan set on some deciding variables. The probable action sequence will be notified on the data of patient given to the conditions of clinical guideline and this will also give critical conditions that need immediate attention. In this project medical grammar rules are applied to extract key decision making steps from the clinical guidelines. In the first step lexical analysis is performed to key- words like 'if this then perform this, all the medical terms will be identified and this extracted rule set will be used to create a XSLT file. The patient data in form of an XML file will be then applied to the XSLT transformations or rule sets to derive final result of action plan specific to that patient.

Keywords : Clinical Data Repository(CDR), Virtual Medical Record(VMR), Abstract Syntax Notation(ASN), Electronic Medical Records(EMR), Medical Logic Modules(MLM), HealthCare Data Dictionary(HDD).

I. Introduction

Data Mining is the science of finding patterns in huge reserves of data, in order to generate useful information from it. Data Mining has potential applications in several fields, one of which is Health Care. The myriad possibilities of improvement in Health Care through Data Mining only further justify the need to apply data mining principles to clinical data. However, prior to applying data mining techniques to garner information from data, the data has to be 'prepared' to ensure the veracity of the information obtained. 'Preparing' the data involves removal of incorrect information or 'noise' from the data and ensuring that the data mining principles are applied on real data. This paper gives a detailed description of the purpose, design and implementation of the Data Mining Framework. The primary purpose of the Data Mining Framework is to help determine trends in patient records to improve Health Care.

Health Care Preparing the data, prior to applying data mining techniques, is a critical step of data mining. It primarily consists of four steps. The first step is the selection of data. This is following by data cleaning, forming the new data and finally formatting the data.

Data has to be selected, prior to applying data mining principles. The data is chosen based on its completeness and correctness. Constraints on the data, such as the data type, could also be factors in the selection of the data.

Author α : Department of Computer Engineering Maharashtra Academy of Engineering Alandi, Pune.

Most of the medical data for the use of doctors are achieved/ available in text form in medical reference books and on Internet in web pages. To derive decision-making information from bulk of data, careful observation of the guideline is required, which tampers quick decision.

Data Mining can automate the process of finding trends and patterns in databases. Large Databases can be analyzed quickly and effectively, using high performance systems and time effective algorithms. As the speed of processing improves, the size of the database can be increased. Also, as the size of the database increases, the accuracy of the predictions also improves. Co-relation among data can be identified.

The data-mining tool, based on the type of task that has to be performed, creates the data model. The modeling phase is an important aspect of the data-mining process. There are a vast number of data-mining techniques that can be applied to the data model. The data modeling technique, which is applied depends on the type of the model. The trends and patterns that are found in the data are based on the data mining technique applied. These patterns can be fed to decision support systems, which can make informed predictions on the data. Some of the more popular data modeling techniques are discussed in the following section

1. Decision tree is an attribute classifier, which takes the form of a tree. It can consist of leaf nodes or internal nodes. The former signifies the value of an attribute while the latter are decision nodes. These nodes, as the name signifies, specify a test that can be performed on the attribute, so that the attributes can be classified as sub-trees. Decision trees algorithms basically use the divide-and-conquer approach to classify. They thus work in a top-down manner, using an attribute at each level split on, that best separates the classes. Decision trees signify rules and are easily interpretable by humans.
2. In Rule Induction methods, rules are created from the data based on statistical values. In this approach, all values in a class are considered. At each stage a rule is generated for these values. Testing the rule under construction with new values creates a rule with maximum accuracy. The value is chosen such that it maximizes the probability of the desired classification.

II. Detailed Clinical Models

While coded vocabularies provide much of the raw material needed to describe clinical information, they are not sufficient alone. If we want to state that a patient has a diagnoisis of breast cancer, we could store the concept, <254837009 | SNOMED-CT | breast cancer>, in her electronic medical record. If we wanted to state that the patient had a family history of breast cancer, we could store the concept, <275862002 | SNOMED-CT | family history of breast cancer>, in her record. Suppose now that we wanted to store the fact that it was the patient's sister that had breast cancer. Currently SNOMED-CT does not have a concept for this. Although a coded vocabulary like SNOMED-CT could add concepts like this, it is not a practical solution. It would require the maintainers of the vocabulary to create concepts for most combinations of disease and family members.

III. Clinical Data Repository

The CDR is a robust electronic medical record system which makes extensive use of coded vocabularies and detailed clinical models.

The detailed clinical models used by the CDR are defined using Abstract Syntax Notation One (ASN.1). ASN.1 is an ISO standard for describing electronic messages [ASN]. As its name implies, ASN.1 provides a syntax for describing messages that is abstract from any specific encoding. However, in addition to the abstract specifica-tion, ASN.1 also defines multiple specifications for specific encodings, including binary and XML encodings. Thanks to its flexibility and efficiency, ASN.1 isused in many dif-ferent areas ranging from telecommunications to genome databases.

The best analogies for understanding what ASN.1 is and how it works are nested structs in the C programming language and XML. All three are tools for defining nested data structures where each field in the structure can have a name and a type. All three tools have distinct concepts for definitions and instances. This means that while instances of C structs are always regions of memory and instances of XML are always text documents, instances of ASN.1 can be represented in many different forms depending on the chosen encoding rules. Since all of the encodings are representationally complete with respect to the abstract model, they are interchangable. Figure 10 gives an example of an ASN.1 definition for a simple detailed clinical model. This figure also illustrates that the type of each item in an ASN.1 definition may be a primitive (e.g. REAL) or it may be the result of another definition (e.g. a CodedConcept).

All coded concepts in the CDR are drawn from IHC's Healthcare Data Diction-ary (HDD), another technology jointly developed by IHC and 3M. The HDD is effectively a large coded vocabulary (over 800,000 concepts with over 4 million synonyms) containing both locally defined concepts and concepts from other coded vocabularies. The names of all the detailed clinical models used in the CDR and the fields they contain are defined as concepts in the HDD. The result is that all data stored in the CDR can be viewed as name-value pairs. The name portion of the pairs is always coded concepts.

The CDR is made up of a database and a set of services that operate on the database. For the most part, data in the database is only accessed through the services. The services perform a couple of functions. First, they provide a common access mechanism to ensure consistent security, auditing, and error handling. Equally important is the way Detailed Clinical Model Table Relational the services handle detailed clinical models. To applications built on the services, the CDR behaves more like an object- oriented database than a relational database. The applications pass instances of detailed clinical models to the services and get other instances of detailed clinical models back. Internally, the data is actually stored in a relational database, but this fact is almost completely hidden from any application.

Although the underlying database is relational, the data is not stored in a traditionally normalized relational manner. Instead the CDR has one table where the services store each instance of a detailed clinical model, formatted as an ASN.1 BER string. Every row in this table has a binary field that holds a BER string. Other fields in each row provide information for indexing purposes such as a patient identifier.

In addition, the services shred the BER strings into another small set of tables. These tables are used for indexing purposes. In effect, all of the data in the CDR is stored twice, once in the BER string and once in the relational tables.

This allows the services to do fast indexed searches in the relational tables to identify the detailed clinical models of interest. They can then read back the entire instance with a single row read instead of the large number of joins it would take to reconstitute the models if they were stored only in a normalized relational format. This is advantageous because applications commonly need the entire detailed clinical models rather than just the information present in a single row of a relational table.

IV. Arden Syntax

The Output from the above step is executable logic in Arden Syntax. Arden Syntax was developed in 1990 as a language for encoding medical knowledge. It was developed in an attempt to address the need to share medical knowledge between hospitals and

other medical institutions. Arden Syntax is currently maintained by the Health Level Seven (HL7) Arden Syntax Special Interest Group and is an ANSI 20 standard. Many vendors of electronic medical records have implemented Arden compilers in their systems.

Arden Syntax is written in units called medical logic modules (MLMs). Each MLM contains the logic necessary for making one medical decision. Portions of an MLM. An Arden Syntax MLM is made up of categories and slots. The three categories in Arden Syntax are the maintenance category, the library category, and the knowledge category. Each category contains a list of slots. The slots in the mainenance category contain information related to knowledge base maintenance and change control.

The maintenance category does not contain any clinical information. The slots in the library category describe the sources of information used in creating the MLM, keywords, and related information. The knowledge category of an MLM is where the clinical logic is represented.

The most significant slots in this category are the data slot and the logic slot. The data slot contains mappings of symbols used in an MLM to data in the target electronic medical record. The logic slot, as its name implies, contains the logic that operates on the data. While Arden Syntax is the best option currently available for sharing medical logic across institutions, it suffers from what is known as the "curly braces problem." Arden Syntax does not specify a notation for referencing data elements in the target electronic medical record (EMR). Rather, such references are written in a form that is understood by the native EMR and placed inside curly braces (e.g. the curly braces may contain a SQL statement specific to a given EMR). This means that while the logic of the module should be portable from one EMR to the next, the references to the data in the EMR are not portable. One proposed solution to the "curly braces problem" is to use and abstraction call the virtual medical record (VMR).

V. Virtual Medical Record

VMR is an abstraction of a data model for a medical record [PRC+04]. It is intended that decision logic can be written against a VMR and then distributed to any number of healthcare organizations, each possibly using a different EMR. Each EMR would have a mapping to the VMR and would therefore be able to translate VMR logic into native queries. In the MLM in Figure 4, curly braces follow the "READ" keyword. In this case the curly braces contain an abbreviated snippet of XML representing a VMR query. The specification of a standard VMR is a current effort of the Clinical Decision Support Technical Committee of HL7 and established it.

The VMR that we use in this project is based on some early work from HL7.This VMR consists of a small set of classes that describe clinically relevant information. These classes include Observation, Substance Administration, and Encounter. Each class has a number of attributes. For example the Observation class has a "code" attribute that specifies the type of the observation, a "value" attribute, and other attributes for capturing information such as the timing and status of the observation. In this project we limit our VMR queries to queries on the code and value attributes of the Observation class. This small subset of the VMR captures a large majority of the information needed to determine clinical trial eligibility.

VI. Extraction and Formula Generation

The first part, Criterion Extraction, takes a web page describing a clinical trial as input. For this thesis we used clinical trials from ClinicalTrials.gov, an internet site sponsored by the National Institutes of Health and the National Library of Medicine. We created a Python script that reads the web page describing a trial and extracts the eligibility criteria as well as available context information. The context information consists of items such as whether a criterion is an inclusion criterion or an exclusion criterion. The output of this part is an XML document containing the criteria and context information. Adapting the system to work with trials from 3 - System Design 24 other sources would involve modifying the Python script to understand the format of the new source.

The second part of Step 1, Formula Generation, takes the XML document with the extracted criteria as an input. This process parses each criterion using a link grammar parser [ST91]. From this it then creates a first order predicate calculus formula representing each criterion as Figure 3 illustrates. This process relies partiallyon recognizable sentence or phrase structure. Since the authors of clinical trials sometimes use telegraphic or ungrammatical phrasing, and since the link grammar parser we are using in this work is not familiar with many medical terms and syntactic constructs, the system is not able to correctly parse some criteria into predicate formulas. 1) The number of constructed rules is equal to or greater than the user-specified threshold. .

The root element of this XML document is labeled "criteria". This element contains a "trial" attribute whose value is the URL of the clinical trial. The "criteria" element contains a sequence of "criterion" elements. Each of these "criterion" elements contains

a sequence of "text" elements followed by a "formula" element. The "text" elements contain text that the system extracts from the trial document. The last "text" element in a sequence contains the eligibility criterion of interest. The preceding "text" elements contain available context information system.

VII. Concept Mapping

The process further takes the XML file described above as input. It attempts to map each criterion to concepts and data structures in the target electronic medical record. For each criterion that is successfully mapped we generate executable code for determining if a patient meets the criterion.CDR (Clinical Data Repository) stores clinical data as instances of detailed clinical models that can be viewed as a series of nested name-value pairs. Recall also that all of the pair names are coded concepts, as are some of the pair values. Since all of these coded concepts are in the HDD (Healthcare Data Dictionary), the mapping task consists largely of trying to match words and phrases from consisting of concepts that are either names or values in the detailed clinical models that are stored in the CDR. The content of the HDD is stored in a normalized relational fashion and we kept the same relational structure in our working subset. This way our system could easily use the live HDD or our subset of the HDD by merely changing a configuration parameter. In addition, we created an abstraction of the HDD for our system with a Terminology Server interface that defines a set of methods for making vocabulary related queries. We then created an implementation of this interface against the HDD.

VIII. Code Generation

Although we have chosen to use Arden Syntax as the language of our executable code, we constructed the code generation subsystem using the same separation of interface and implementation that we used in other areas. Therefore generating code in a different language would only require the interested party to supply an appropriate implementation of the generator interface.

Using metadata from the target EMR, the system determines that "heart disease" is valid in the value part of a name-value pair. Thus, for instance - A sample Arden Syntax read statement containing a VMR query. "VMR Query" element contains a "value" element. If the mapped concept serves as the name part of a pair, then a "code" element replaces a "value" element in the query. The valid values of this attribute depend on the type of the element that is contained within the "value" element. In this case the "value" element contains a "cd" element representing a coded concept. The comparison operations that are valid for a coded concept include "equals" and "isa." If the contents of the "value" element represented a numeric value, then numeric comparison operators such as "equals," "less than," and "greater than" would be applicable.

The second step in generating code to determine eligibility takes place after all of the criteria have been considered for mapping. In this step we generate the Arden Syntax MLM. The MLM we generate is focused on the executable logic. Even though the vast majority of slots in an MLM are required by the specification, only a handful are useful for machine execution. Most of the remaining slots are intended for human perusal. Therefore, for this project we populate only the small number of slots that are useful for automated rocessing. We do not generate any slots in the maintenance category. In the library category we populate the inks slot with the URL of original clinical trial.

IX. Conclusion

Clinical medicine is one of the most interesting areas in which data mining may have an important practical impact. The widespread availability of large clinical data collections enables thorough retrospective analysis, which may give healthcare institutions an unprecedented opportunity to better understand the nature and peculiarity of the undergoing clinical processes.

Future Scope

Combine the whole process of clinical process with computer generated treatment recommendations.

Fast and Efficient implementation of clinical guideline reference for the medical practitioners.

Dynamic updating of clinical guidelines according to trial results.

References Références Referencias

1. Rafael S. Parpinelli, Heitor S. Lopes, Member, IEEE, and Alex A. Freitas.
2. Predictive data mining in clinical medicine: a focus on selected methods and Applications Riccardo Bellazzi, Fulvia, Ferrazzi and Lucia Sacchi.
3. Predictive data mining in clinical medicine: Current issues and guidelines by Riccardo Bellazzia,, Blaz Zupanb
4. Data Mining Framework by Hemambika Payyappillil, College of Engineering and Mineral Resources at West Virginia University.