Chapter

Using Artificial Intelligence and Big Data-Based Documents to Optimize Medical Coding

Joseph Noussa-Yao, Didier Heudes and Patrice Degoulet

Abstract

Clinical information systems (CISs) in some hospitals streamline the data management from data warehouses. These warehouses contain heterogeneous information from all medical specialties that offer patient care services. It is increasingly difficult to manage large volumes of data in a specific clinical context such as quality coding of medical services. The document-based not only SQL (NoSQL) model can provide an accessible, extensive, and robust coding data management framework while maintaining certain flexibility. This paper focuses on the design and implementation of a big data-coding warehouse, and it also defines the rules to convert a conceptual model of coding into a document-oriented logical model. Using that model, we implemented and analyzed a big data-coding warehouse via the MongoDB database and evaluated it using data research mono- and multi-criteria and then calculated the precision of our model.

Keywords: diagnostic coding, clinical decision support, decision-making, big data, optimization, medical diagnostic computing

1. Introduction

Care processes are becoming increasingly complex with the growth of technological, biological, and genetic knowledge [1]. This leads naturally to a subdivision of medical specialties, with increasing patient's care costs. Such subdivision, in view of the multiplicity of pathologies of certain patients, complicates diagnosis coding measures, regardless of the coding plan used in clinical and medical-economic settings. It was assumed that incorrect coding of medical information caused, on average, a 14.7% hospital revenue loss per patient [2]. More than three-quarters of these errors are caused by clinicians. These numbers are explained by the intricacy of the classifications and nomenclatures used to code medical acts [3]. In addition, they make it difficult for medical officers to understand the process involved in the coding of this activity [4]. In this context, the establishment of clinical information systems in hospitals can be a key factor in optimizing the coding process of medical information and therefore the expenses of healthcare. The successful establishment of the latter is not done without challenge. In fact, its deployment is based on the data warehouse, which plays an important role in the collection and analysis of large volume of data for decision support. Generally, a data warehouse is often implanted under the relational database management systems (DBMS). The latter obtrudes itself by the richness of their functionality and performance of their

request. Nevertheless, they are inadequate to build distributed data warehouses and needful to cope with the scalability of storage space and the increase of hospital stay data [5]. In addition, execution of decision requests demeans the performance of data warehouses in DBMS [6]. In the context of the Georges Pompidou European Hospital, the data warehouse is rapidly growing and contains structured (ICD-10 code, etc.) and unstructured (natural language text, etc.) data of stays from different medical specialties. These data are scattered and do not offer direct access to the medical act coding data based on hierarchies of the 10th revision of International Classification of Disease (ICD-10) for diagnostics as well as the French Common Classification of Medical Act (CCMA) for medical procedures. Moreover, the complexity of the classification in a coding process poses a major problem for sub coding (code forgotten) and over coding (addition of codes not justified) of hospital stays [7]. These challenges can be addressed by providing physicians with a big data storage environment dedicated to coding hospital stays whose particularity is to combine their size (volume), frequency of updates (velocity), or diversity (variety) [8]. Volume, variety, and velocity, often referred to as the three Vs, capture the real meaning of big data [9].

Because big data bring many attractive opportunities to knowledge management [9, 10], the aim of this study is to model the coding data of hospital stays extracted from a data warehouse and implement them in a document-oriented NoSQL data model capable of storing a large distributed big data set. This study sought also to design a big data-coding warehouse efficient for medical coding according to a NoSQL document-oriented data model, which will allow to obtain the optimal combination of codes (diagnoses and acts) for any given reason for care.

2. Methods

This section describes the methodology used to design, implement, and evaluate our data model. Generally, there is a need for a semantic data model to define how data will be structured and related in the database [11], and it is generally acquired that Unified Modeling Language (UML) meets this requirement [12]. Therefore, we first used the formalism of UML [13, 14] to build a conceptual model describing big data of hospital stays. Then, the corresponding rules were used to convert the conceptual model to NoSQL database. In this paper, we choose to focus on document-oriented NoSQL model, namely, MongoDB. This model developed since 2007 by the company of the same name is considered to be the most efficient in terms of performance, for multi-criteria access queries. Finally, the document-oriented model of the big data warehouse was described in JavaScript Object Notation (JSON) format and implemented in the MongoDB database. Subsequently, decision requests were used to evaluate the model.

2.1 Data source

The CIS grouping software for Georges Pompidou European Hospital was used to extract the base of hospital summary report (HSR) from the digestive, oncologic, and orthopedic surgery units. This base of HSR contains a year of hospital stays coded and validated by the department of medical information. Each HSR has medical benefit entities and temporal and administrative patients' data. The entity of medical benefits represented by the medical act and diagnostics appears in three different types of diagnoses: the associated significant diagnosis (ASD), principal diagnosis (PD), and related diagnosis (RD). Moreover, there are various types of acts such as a surgery act and medical technical act.

2.2 Conceptual data model

The goal of a document-oriented database is the representation of more or less complex information that satisfies the needs of flexibility, richness of structure, etc. Modeling of a big data-coding warehouse is a function of the hospital stay's structuring elements. The hospital stay is a document represented by a pair (key, value) and has a tree-shaped structure. The stay entity is the root of the tree. The entities (key) and values for coming up with a conceptual model (**Figure 1**) were designed from the base of HSR. The main entities were defined as an object class that consisted of stay, patient, movement of the patient between the different clinical unit entities, and terminology. The medical benefit entity is a sub-document of stay entities in which the related act and diagnosis are sub-entities.

The entities have a heterogeneous structure and multiple values. The relationship between them is of cardinality "n,m." To ensure that they are unified, it is important to have a flexible open data schema whose structure can be extensible and is able to adapt to more or less important variations. The rank of medical acts, as well as the rank of diagnoses, can easily complete the corresponding type of diagnosis and acts. The rule used to convert the conceptual model to a document-based model is based on the tree model so that the code associates the structure (the tree) and the content (the text in the sheets).

2.3 Correspondence rules with the conceptual/logical level

The coding data are arranged in rows and columns through the big data warehouse model. It is structured into the nested documents in document-oriented NoSQL. The entity stay is a set of facts (patient, movement, medical benefit, etc.) where an instance turns into a nested document. Every sub-entity (movement, medical benefit, etc.) is changed into a nested document. Every sub-sub-entity (Diag, Act, etc.) is changed into a nested document. Every entity also changes to a nested document held in the same document as the fact instance. A stay that has only one main diagnosis is converted into a document that is turned into only one sub-entity Diag. The attributes (key, values) of the other sub-entities are null. There is no preservation of hierarchical organization.

2.4 Document-based model

Under the prism of rules published in Section 2.3, the document-oriented model considers each hospital stay as a key associated with a value. The values can be either

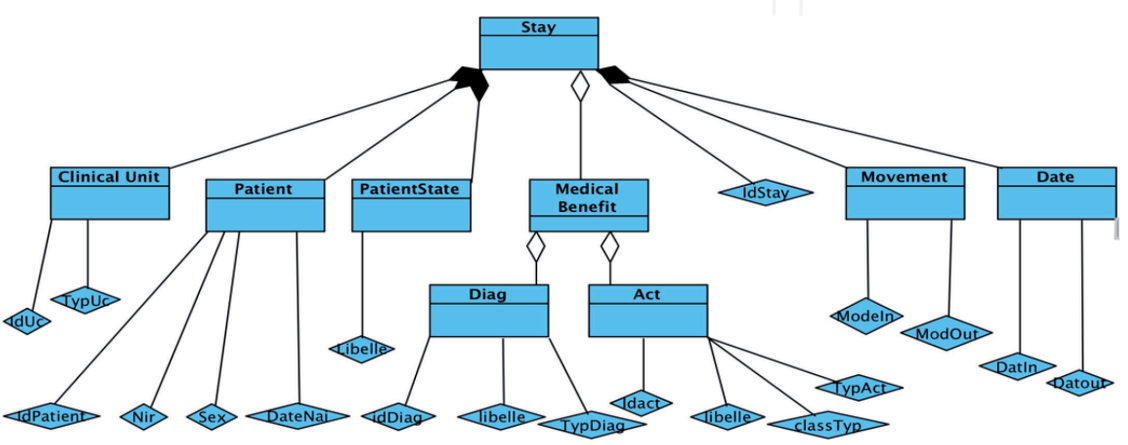

Figure 1.
Hospital stay conceptual data model.

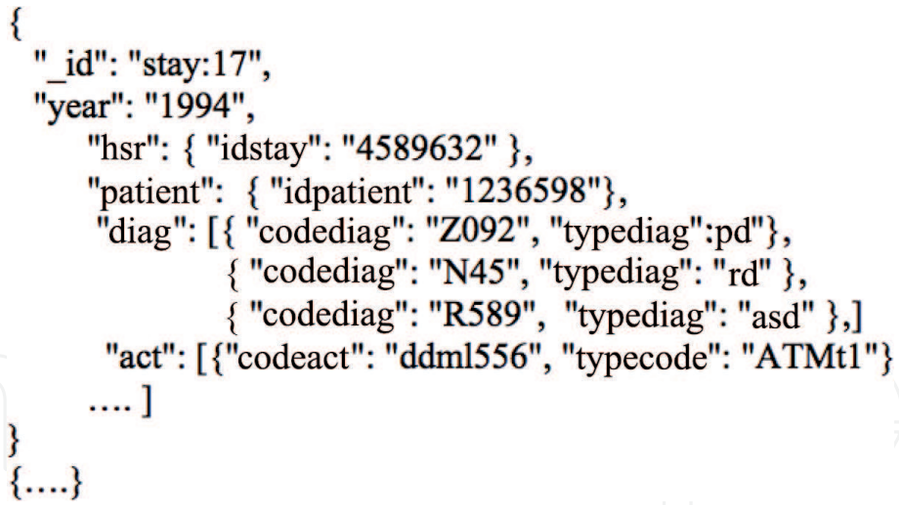

Figure 2.
Document-based schema.

atomic value (date of stay) or other documents (patient, medical benefit, etc.). The base of HSRs is a collection of hospital stays, and every stay corresponds to a single entry of this base. The collection includes documents for certain entities, shown through their structure, such as name of the key (medical unit, patients, etc.) and its content (values of the keys (integer (id stay), string (patients, ICD-10), etc.)). The rest of the entities are shown through structural imbrications, such as the medical benefit that is a medical act value aggregate. The entities "medical benefits" refer to an aggregate of a key (diag), which is also an aggregate of (value is ICD-10 code, key is codediag). A document can be defined as a hierarchy of elements that can act as atomic values or nested documents shown by a new set of pairs (value, attribute). There is a simple attribute in the HSR, which makes these values to atomic. The values of compounded attributes are nested documents, as shown in **Figure 2**.

The relationship between different entities is translated in the form of nesting. A document model uses the specific NoSQL request language to query in width and depth all entities present in the collection. A multi- and mono-criteria request can be carried out. An example could be as follows: for a given diagnostic and the type of diagnostic, give all associated diagnostic codes and act codes.

2.5 Evaluation

To evaluate our model, framework MongoDB was deployed in only one data node. The system resources used were (8Go RAM, processor i5–4 heats 2 Terabytes hard disk). MongoDB is a key-value system using document-oriented storage. The volume of data received by one node for the test is 1.6 million documents representing 1 year of the hospital stays. The volume of documents can be worm at 40 times the initial volume. These data are divided into two major groups: encoded and rejected data. The interest of the rejected data is to expand the database, introduce the noise, and have a case of associations of diagnostic code to avoid. Two evaluations were performed. The first one was performed by calculating the model performance (multi-criteria request (Query #2), mono's (Query #1) elapsed time), and the second one was performed by calculating the precision and recall of the model. The initial step was to create two main groups of requests arranged by dimensionality and selectivity validated by the business process, which can be used in a real context. Dimensionality is the value of different keys of the entities ("typediag" and/or "typeact"). Selectivity refers to the degree of data elimination through an aggregate function on the search attribute (code = "CCMA code or ICD-10 code").

Query #1 is expressed as follows: for a given diagnosis or act codes, find all associated code and their corresponding type (typediag, typeact). Three functional use cases were used. The first one concerns the specific case of Z codes as part of the entire ICD-10 code set. Z codes are diagnosis codes used for situations where patients Do not have a known disorder and required a related code to precise the real medical. The second one concerns the code of another chapter of ICD-10, and the last concerns the CCMA act code. Query #2 is expressed as follows: for a given diagnosis and act codes, find all associated codes and their corresponding type (typediag, typeact). Second, through cross-validation, random evaluation is used in measuring the impact of a distributed data storage device on medical procedures' coding optimization. This method provides an opportunity for inserting the process of selection codes of medical acts in a random draw. There are two groups that arise from this draw: an oriented test group with 20% sampling size and 50% of the volume of the data warehouse. The second is a control group with a 50% sampling size and 80% of the data warehouse volume of coding derived by the medical information department. Ten samplings with the same percentage were generated to perform the tests. We used the request previously described to compute the precision and recall of the model. Precision is the ratio between the number of correct associations and the total number of associations, and the ratio is the number of correct associations to the number of all associations to be corrected.

3. Results

The document-oriented model of the big data-coding warehouse was implemented in the MongoDB database. Ten separate single-criteria queries were executed with an elapsed time between 75 and 90 milliseconds (ms), while an elapsed time between 80 and 110 ms was obtained during the execution of 10 separate multi-criteria queries.

Query #1 requests the data warehouse to display all association codes, in which the diagnosis code is equal to "Z092" in the ICD-10 coding system, corresponding to "the pharmacotherapy for other conditions." Associated codes obtained are the associated diagnostic code "E780" used to code "pure hypercholesterolemia" and the act code "EBQM002" used to code "Doppler ultrasonography of extracranial cervicocephalic arteries, with Doppler ultrasound of lower extremity arteries." The elapsed time of this request is 90 ms (**Table 1**).

Table 1 presents results of five sequences of request of Query #1 where requested code represents the code to be queried, associated code represents the obtained associated codes, typology represents different types of diagnostic, and elapsed time represents the execution time of request.

Query #2 requests the data warehouse to display all association codes, in which the type of diagnosis is the main diagnosis and the diagnosis code is equal to "I51.4" in the ICD-10 coding system, corresponding to "cardiomegaly." The response time obtained with no index is approximately 1900 ms. The response time obtained with a diagnostic code indexed is approximately 110 ms. The associated codes are "I080" corresponding to "disorders of mitral and aortic valves" and "D721" corresponding to "eosinophilia," and associated act is "DEQP003" corresponding to "electrocardiography at least 12 leads" (**Table 2**).

Table 2 presents results of five sequences of request of Query #2 where requested code represents the code to be queried and its typologies, associated code represents obtained associated codes (diagnostic and act), typology represents different types of diagnostic, and elapsed time represents the execution time of request.

Requested code	Associated code	Typology	Elapsed time (ms)
I49.9	Not associated code	No typology	75
Z09.8	D35.0	(PD, RD) or (PD, ASD)	85
Z09.8	EBQM002, E78.0	PD, RD, ASD	90
E660.0	I10, N17.9	PD, RD, ASD	87
DEQP003	Z864, I708, E70.8	ACT, PD, RD, ASD	90

Table 1.
Associations of diagnosis codes according to their typology and their elapsed time.

Requested code/typology	Associated code	Typology	Elapsed time (ms)
I49.9/dp	Not associated code	No typology	75
Z09.8/dp	Q21.3	(PD, RD) or (PD, ASD)	85
(Z09.8/dp) and (N185/das)	Z992.1, D638, and JVJB001	PD, RD, and ASD	89
E26.0/dr	Z71.3, D35.0, and DZQM006	PD, RD, ASD, and/or ACT	87
EQQP008	N18.5, I70.2, and Z098	ACT, PD, RD, and ASD	

Table 2.
Associations of diagnosis codes according to their typology and their elapsed time.

Learn/control DB (%)	Precision (%)	Recall (%)
Mono-criteria 50/50	40	25
Mono-criteria 80/20	92	87
Multi-criteria 80/20	80	70

Table 3.
Evaluation results of the big data model.

The list of associated codes present in **Tables 1** and **2** is not exhaustive; it can be extended to more than 100. We make the choice to present a small number.

These results show that the main coding rules have been respected. The associated diagnosis must always be coupled to Z code declared as the main diagnostic and associated acts linked to disease declared as main diagnostic. The presence of related diagnostic demonstrates the quality of associated code containing in the data warehouse.

Query #1 and Query #2 were used to compute the precision and the robustness of the model (**Table 3**).

Based on the observation, the least selective (more lines selected) queries required a long execution time. According to our evaluation, we observed that the system is bijective and corresponds to the reality of the coding of clinical activity of HEGP. This suggests that we can, from the document-oriented model, recover the initial encoding data and vice versa. In this regard, it is apparent that everything that has been set in the big data warehouse corresponds to the reality of the patient. The data warehouse gives the possibility of being more aware of the coding performed in the previous year.

Based on the requests defined above and executed using the learn/control database, **Table 1** shows the results of the evaluation provided by the big data

model. The 50/50% precision test was 40 and 25% for recall versus 92/87% for the mono-criteria (Query #1) request. For the multi-criteria (Query #2) request, it was 80/70% for the 80/20% test. Although there are some errors in the test, the sensitivity of conformity computation was 0.8, and its specification was 0.7. Based on this result, the level of accuracy depended on the number of associated diagnostic codes present in the association of codes.

4. Discussion and conclusion

This study investigated the process for implementation of a big data-coding warehouse for coding support in a document-oriented NoSQL system. We observed that flexibility is the particularity of this model as it allows inserting redundancy into the database. A stay with four ASD codes and one PD code is split into four documents. The duplicated line is high when there are more associated diagnoses and medical acts. Therefore, presenting one entity is easier in the entire document. The case of "stay" with only a primary diagnosis, one or more associated diagnosis, and/or without a medical act can be easily inserted in the database without the need to implement a generic code to replace the missing one. In most cases, the addition of a generic code is meant to let the physician understand that there is no need of associating a diagnostic code used with the medical act. This system is advantageous since there is complete information because the issue of missing data is solved. Therefore, the information can be handled without any need to join. Only one reading is needed to get all information. If there is no link between the documents, it is possible to arrange the collection without any challenge. This is an essential part of the construction of a big data-coding warehouse. However, one of the disadvantages associated with this model is that the hierarchization of access does not allow access to ICD-10 code information without going through the type of medical benefit, in addition to the redundancy; there are two pseudorandom choices that provide effective results, while the hazardous choice (50/50%) produces wrong results. To generate huge volumes of data, we used the same "HSR base" and swapped the name ICD-10 by the concept "Obicd10'' and CCMA by the concept "Obccam" (Ob as rejected). The rejected data were used to show that, in the optimization process of coding, we learn about as many accepted cases as rejected cases. The major interest in building the coding aid data warehouse is to use the huge volumes of coding information from a large number of hospitals because it is more exhaustive. The model that was implemented allows obtaining an optimal combination of codes (diagnosis, acts) for a given reason for care. Because of the way they are structured, relational databases usually scale vertically—a single server has to host the entire database to ensure reliability and continuous availability of data. This gets expensive quickly, places limits on scale, and creates a relatively small number of failure points for database infrastructure. It's why we propose our model to solve this problem. Indeed, our coding aid data warehouse scales horizontally—several servers host the entire database, allow grouping of all the relevant data for the diagnosis and medical coding in a generic way, to enrich the coding data by crossing the coding information from other hospital sources and to allow for easier exploration of the coding code associations. It's a system that is subject to expertise. This fact Does not remove the richness of Clinical Data Warehouse (CDW). Our contribution consists of building a specific CDW-based document to propose an "in silico" test framework to enhance the efficacy of algorithms used to optimize coding as an example of algorithm based on manual decision-making paper [15] and various natural language processing (NLP) tools associated with the EHR in−/outpatient summary reports [16].

Author details

Joseph Noussa-Yao[1,2*], Didier Heudes[1,3] and Patrice Degoulet[1,3]

1 INSERM, UMR_S 1138, Paris, France

2 Pierre and Marie Curie University, Paris, France

3 European Hospital Georges Pompidou, France

*Address all correspondence to: jnoussa@gmail.com

References

[1] Ralston JD, Larson EB. Crossing to safety: T ransforming healthcare organizations for patient safety. Journal of Postgraduate Medicine. 2005;**51**(1):61-67

[2] Nouraei SA, O'Hanlon S, Butler CR, Hadovsky A, Donald E, Benjamin E, et al. A multidisciplinary audit of clinical coding accuracy in otolaryngology: Financial, managerial and clinical governance considerations under payment-by-results. Clinical Otolaryngology. 2009;**34**(3):259-260

[3] O'Malley KJ, Cook KF, et al. Measuring diagnoses: ICD code accuracy. Health Services Research. 2005;**40**(5 Part II):1620-1639. DOI: 10.1111/j.1475-6773.2005.00444.x

[4] Puentes J, Montagner J, Lecornu L, Cauvin JM. Information quality measurement of medical encoding support based on usability. Computer Methods and Programs in Biomedicine. 2013;**112**:329-342

[5] Leavitt N. Will nosql database live up to their promise? Computer. 2010;**43**(2):12-14

[6] Golfarelli M, Maio D, Rizzi S. The dimensional fact model: A conceptual model for data warehouses. International Journal of Cooperative Information Systems. 1998;**7**:215-247

[7] Chaudhuri S, Dayal U. An overview of data warehousing and OLAP technology. ACM Sigmod Record. 1997;**26**:65-74

[8] Hay SI, George DB, Moyes CL, Brownstein JS. Big Data opportunities for global infectious disease surveillance. PLoS Medicine. 2013;**10**(4):e1001413. DOI: 10.1371/journal.pmed.1001413

[9] Chen CP, Zhang CY. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences. 2014;**275**:314-347. DOI:10.1016/j.ins.2014.01.015

[10] Fredriksson C. Knowledge management with big-data creating new possibilities for organizations. In: The XXIVth Nordic Local Government Research Conference (NORKOM); Gothenburg: Nordiska kommunforskarkonferensen; 2015

[11] Daniel G, Sunyé G, Cabot J. UML to graph DB: Mapping conceptual schemas to graph databases. In: Comyn-Wattiau I, Tanaka K, Song IY, Yamamoto S, Saeki M. editors. Conceptual Modeling. Vol. 9974. Springer, Cham.: Lecture Notes in Computer Science; 2016. https://doi.org/10.1007/978-3-319-46397-1_33

[12] Abadi D, Boncz P, Harizopoulos S, Idreos S, Madden S. The design and implementation of modern column-oriented database systems. Foundations and Trends in Databases. 2013;**5**(3):197-280. DOI: 10.1561/1900000024

[13] Object Management Group, Inc. 2005. UML 2.0 Superstructure. Available from: http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04.pdf

[14] Luj'an-Mora S, Trujillo J, Song IY. A UML profile for multidimensional modeling in data warehouses. Data and Knowledge Engineering. 2006;**59**(3):725-769

[15] Lecomu L, Thillay G, Le Guillou C, Garreau PJ, Saliou P, Puentes J, et al. REFEROCOD: A probabilistic method to medical coding support. In: Conference Proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2009; 2009. pp. 3421-3424

[16] Mitchell JB, Bubolz T, Paul JE, Pashos CL, Escarce JJ, Muhlbaier LH, et al. Using medicare claims for outcomes research. Medical Care. 1994;**32**(7 Suppl):JS38-JS51