Using a Snowflake Data Model and Autocompletion to Support Diagnostic Coding in Acute Care Hospitals

Joseph NOUSSA-YAO\textsuperscript{a,b,1}, Abdelali BOUSSADI\textsuperscript{a,c}, Monique RICHARD\textsuperscript{c}, Didier HEUDES\textsuperscript{a,c}, and Patrice DEGOULET\textsuperscript{a,c}

\textsuperscript{a}INSERM, UMR S 1138, CRC, Team 22, Paris, France
\textsuperscript{b}Pierre and Marie Curie University, Paris, France
\textsuperscript{c}Hôpital Européen Georges Pompidou (HEGP), Paris, France

Abstract: Purpose: Efficient and adequate coding is essential for all hospitals to optimize funding, follow activity, and perform epidemiological studies. Objective: We propose an autocompletion method for optimizing diagnostic coding in acute care hospitals. Methods: Using a terminology snowflake model integrating SNOMED 3.5 and ICD-10 codes, autocompletion algorithms generate a list of diagnostic expressions from partial input concepts. Results: A general autocompletion component has been developed and tested on a set of inpatient summary reports. Concepts expressed as strings of three or four characters return a noisy list of diagnostic labels or codes. Concepts expressed as groups of strings return lists that are semantically close to the labels present in hospital reports. The most pertinent information lies in the length of the expressions entered. Conclusion: Autocompletion can be a complementary tool to existing coding support systems.

Keywords. Diagnostic coding, Optimization, Autocomplete, ICD-10 classification

Introduction

Coding clinical procedures and diagnoses from hospital discharge summaries is a critical activity within current hospital information systems because this piece of information is used in many countries to support hospital funding and to monitor the quality of healthcare \cite{1,2}. Finally, hospital discharge summaries and their associated codes are an important source of population-based data used in epidemiologic studies \cite{3}. However, the reporting of diagnosis and procedure codes is a complicated and time-consuming task for physicians. Hospitals sometimes employ expert coders, leading to additional spending with mixed outcomes. Expert coders without any particular knowledge of patient histories can produce lists of codes that are considered to be specific but not sensitive enough \cite{4}. Hence, coding mistakes are commonly accompanied by a risk of a posteriori negative budget adjustments \cite{1}. Well-recognized concerns regarding diagnosis and activity codes and their clinical meaningfulness, timeliness and accuracy have been reported in the scientific literature \cite{5,6}. Despite the

1 Corresponding Author: J. Noussa Yao, Inserm-UMRS 1138, Team 22, 15 rue de l’Ecole de Médecine, 75006 Paris, France.
fact that more recent studies report improvements in using these data in epidemiologic studies thanks to the growing availability of electronic health records (EHR) and clinical data warehouses, researchers and physicians continue to call for a better handling of diagnosis and activity coding in hospitals. Different coding support strategies have been investigated [7]. They include manual decision-making paper-based algorithms, paper/computer-based decision trees, and various natural language processing (NLP) tools associated with the EHR in/outpatient summary reports [5]. Other studies have used probabilistic methods (e.g., naive Bayesian modeling, k-nearest neighbors algorithms, vector support machines) to optimize the coding of acts and diagnoses [1]. In hospitals that use an integrated clinical information system (CIS), codes can be partially generated from the EHR.

We describe in this work a complementary autocompletion approach that is commonly used on the World Wide Web for searching and mining information. Starting from partial diagnostic or activity names, lists of potential diagnostic and activity expressions and their related codes can be proposed to the coding actor.

1. Methods

Terminology modeling: We designed and implemented a snowflake data model to handle and use ICD-10 classification and SNOMED 3.5 nomenclature combined with a database of coding rules provided by the French Technical Agency for Hospitalization Information regarding the degree of severity for a set of diagnostic codes. In the snowflake terminology model developed for the autocompletion procedure (Figure 1), the “code” is a key entity that represents diagnostic facts. Entities such as exclusions, inclusions, valorizations, code rules, “daget” codes, medical summary units and correspondence tables are dimensions and thus, the axes on which analyses are carried out. The snowflake schema allows the terms of the ICD-10 classification to be combined with those of the SNOMED nomenclature and includes the possibility of adding levels of severity to assigned ICD-10 diagnostic concepts. This combination makes it possible to respect the structure of each terminology and bring together the associated metadata. The structures of the terminologies are linked to dedicated points in the schema. This makes it simple to integrate specific updates into the schema. It also makes it possible to use the correspondence between ICD-10 codes and SNOMED not only to resolve synonymy problems (e.g., sugar diabetes and sugar disease), homonymy problems (e.g., cortical atrophy of the brain and cortical atrophy of the kidney), and hyperonymy problems but also to constitute simple dynamic queries to generate an exhaustive list of diagnostic expressions candidates.

![Figure 1. Autocompletion terminology model](image-url)
**Autocompletion:** Autocompletion algorithms are based on the QAC (query auto completion) method [8] and the general query model that we constructed. We search for a label, code or severity in the database such that the diagnostic concepts contain, or begin with, or end with the input characters. When the prefix of a medical concept is input, a database query is carried out, and the QAC algorithm automatically returns a list of concepts or diagnostic expressions. The length of the list depends on the length of the medical concept entered. This list consists of the diagnostic codes described both in the ICD-10 classification and the SNOMED nomenclature, with the corresponding code and the associated degree of severity. Autocompletion thus provides a means of reducing the large range of diagnostic expressions present in the ICD-10 to the expressions closest to the diagnosis retained during patient management. We then construct the three key functions of this approach: 1- dynamically supplies the most relevant terms to the coding actor; 2- produces a minimal list of responses containing candidate expressions; 3- classifies the results according to the ICD-10 hierarchy. Thus, when a particular label is selected, a code and its valorization level for the years N and N-1 are automatically supplied. We then considered the question of the provision of additional information combining the exclusion and inclusion labels and group delimitation. Exclusion labels are misleading (or “false friends”) when they are semantically close to the label considered but should be classified elsewhere. By contrast, inclusion labels are ones classified in the same rubric as the candidate labels. For example, cerebrotendinous cholesterosis, which is coded E75.5, is a false friend of mixed hyperlipidemia (code E78.2), whereas hypercholesterolemia and endogenous hyperglyceridemia, a tuberous xanthoma, are truly related to mixed cholesterolemia, all these conditions bearing the code E78.2.

**Dataset:** A set of hospital summary reports from digestive and orthopedic surgery and oncology was extracted from the EHR of the Hôpital Européen Georges Pompidou CIS. These reports contain diagnoses written in natural language. Each diagnosis is composed of a set of medical acronyms and correctly and incorrectly formulated medical concepts, diagnosis expressions (including diacritic characters), and acronyms of groups of expressions separated by commas or diagnostic concepts linked by generalization/specialization relationships. The tool was tested by simulating several types of diagnostic expression inputs linked to three different cases: 1- Single words, 2- Searches made with concepts consisting of several words, 3- Compound terms including syntax errors.

2. Results

A full autocompletion tool was implemented in Java, Javascript and PHP and integrated with the MySQL terminology database. The system has been deployed as a Web service and includes three components: The first component consists of a user interface and themes for managing text entries; the second component is constituted by a cache process whose function is to directly enable the provision of a diagnostic list without the need to make a call to the service engine; the third component is a service engine that includes three Web services that access the data source to respond to queries and access an online referral service that will call the three previous Web services and load the adequate service to update the data source. Figure 2 illustrates a
sequence diagram for the autocompletion component. A physician enters the first characters of a concept being coded. A list containing all diagnoses and associated ICD-10 codes containing these characters is returned to the user. At each step of concept entry, the list is reduced and offers diagnoses closer to the one considered. At the end of the entry, a list of more precise diagnoses is proposed. The output expression produced includes diagnostic concepts consisting of a code associated with a single label, one or several implicit or explicit exclusions, one or several implicit or explicit inclusions, relationships to signs, causality and exclusion relations.

![Sequence diagram for autocompletion component](image)

**Figure 2. Use case of the autocompletion web service**

Table 1 describes the results of an evaluation of the autocompletion tool in different situations: a single word concept (e.g., artery) and a two-word concept (e.g., pulmonary artery). For strings of three characters, the search generated in this example corresponds to 1924 labels versus only 186 for strings of four characters. For searches based on two strings separated by a space, the tool generates “noisy” lists containing hundreds of labels including diseases relating to different systems. In contrast, when the entire phrase is input, the list of labels obtained is semantically close to the labels present in the hospital report.

<table>
<thead>
<tr>
<th>Concepts (single word)</th>
<th>Containing the string “%art%”</th>
<th>Containing the string “%arte%”</th>
<th>Beginning with “art%”</th>
<th>Beginning with “arte%”</th>
<th>Beginning with complete concept “artere%”</th>
<th>Containing complete concept “artere%”</th>
</tr>
</thead>
<tbody>
<tr>
<td>list size</td>
<td>1924</td>
<td>186</td>
<td>456</td>
<td>25</td>
<td>15</td>
<td>119</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts (several words)</th>
<th>Containing “%pul% %art%”</th>
<th>Beginning “%pulm% art%”</th>
<th>Beginning “%pulm% arte%”</th>
<th>Containing “%Pulmonary artery%”</th>
<th>Containing “%Pulmonary artery%”</th>
<th>Beginning “Pulmonary artery%”</th>
</tr>
</thead>
<tbody>
<tr>
<td>list size</td>
<td>100</td>
<td>12</td>
<td>2</td>
<td>12</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts (syntax error)</th>
<th>Containing “%pul% %art%”</th>
<th>Containing “%pulm% %art%”</th>
<th>Containing “%Pulmonary artery%”</th>
<th>Containing “%Pulmonary artery%”</th>
</tr>
</thead>
<tbody>
<tr>
<td>list size</td>
<td>100</td>
<td>12</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

The string “art%” generated a list of all the concepts beginning with “art%” (e.g., artery, arthrosis, articular), whereas “%art%” generated a list of all the concepts containing the character string “art” (e.g., artery, arthrosis, articular, arterial atresia, arterial stenosis)(Table 1). The tool generates a “noisy” list containing 100 labels.
including diseases relating to two systems (vascular and pulmonary). Concepts such as “pulmonary artery aneurysm” I28.1 code and “pulmonary artery atresia” Q255 code are included in the list. When parasite characters are intentionally included in the diagnostic (capital letters, accents etc.), the tool yields all the codes corresponding to the label, the related codes from the ICD10 hierarchy and the codes included in other chapters of the ICD10. We also tested the autocompletion tool with generalization/specialization relationships. Too general concepts obviously generate extensive lists with potentially higher levels of background noise. For example a search for the concept “acute myocardial infarction” generates 16 diagnostic expressions, whereas a search for the more general concept “myocardial infarction” generates a list of 24 diagnostic expressions. The autocompletion tool is not sensitive to cedillas, upper and lower case or accentuated characters and does not generate polysemy unless other words are added.

3. Discussion and conclusion

The list of results obtained does not contain false friends. The more precise the character string is initially entered, the less background noise is likely to be found in the list obtained which means that a balance need to be found between the number of characters entered and the lists provided as output. Current limitation of the autocompletion tools include a knowledge of the principles of organization of the classifications and nomenclatures used as well as the coding rules that can be specific to each country. ICD10, the reference classification for the coding used for DRG coding in many countries, contains imprecise labels that do not correspond to diagnosis made during patient management, a situation inherent to the classification and not to the autocompletion approach. We tried to overcome such synonymy problems by reconstituting non-exhaustive list of synonyms, which were incorporated into the snowflake model.

References