Internal Structure of a Disease Name and its Application for ICD Coding

Emiko Yamada\textsuperscript{a}, Eiji Aramaki\textsuperscript{b}, Takeshi Imai\textsuperscript{c}, Kazuhiko Ohe\textsuperscript{a}

\textsuperscript{a} Department of Medical Informatics, Graduate School of Medicine, The University of Tokyo, Japan
\textsuperscript{b} Center for Knowledge Structuring, University of Tokyo, Japan
\textsuperscript{c} Center for Disease Biology and Integrative Medicine, Faculty of Medicine, The University of Tokyo, Japan

Abstract

ICD-coding is a complex and difficult task. Coding results vary a great deal depending on each coder’s ability. Although the Japanese Standard Disease-Code Master facilitates the coding tasks, it also engenders post-coordination problems derived from combinations of basic diseases (with ICD code) and modifiers. Post-coordination sometimes alters the original ICD code dramatically. To solve this problem, this paper presents a proposal for using internal structures of disease names to correct the ICD code. First, we built an internal structure analyzer, which achieved high (83.7\%) accuracy. Results demonstrated that the analyzed output is helpful for precise ICD-coding tasks.

Keyword:

ICD-10, Disease name, Multi-word expression, Internal structure, Dependency analysis, Post-coordination

Introduction

The international standard diagnostic classification for all general epidemiological purposes and many health management purposes is ICD-10. It is used for analyses of general health circumstances of population groups and monitoring of the incidence and prevalence of diseases [1]. However selecting a suitable disease code for each patient requires a high level of understanding of both patient data and medical knowledge.

We classified the complex language phenomena in ICD coding into three types:

1. Spelling variation
   
   Myo\textsubscript{cardial} infarction / Myo\textsubscript{kardial} infarction [I21.9]  
   
   (value in brackets: ICD code)

   The first type, spelling variation, mostly originates from transliteration problems, which occurs in vocabulary importing. Because this phenomenon is a general research topic, and because various methods have been proposed [2,3], this paper does not address this problem.

2. Synonym / Hypernym
   
   Coats’ disease / exudative retinitis [H35.0]

   The second type, synonym/hypernym, requires extra knowledge such as ontology. It is also beyond this paper’s scope.

3. Order / Existence of Modifier (Post coordination)
   
   femoral fracture / femoral incomplete fracture [S72.9]  
   femoral shaft fracture [S72.3]

   The third type—the target of this paper—is post coordination, which is the combination of a disease whose correct ICD code is known and modifiers. Post coordination must cope with the expansive variety of disease names. Moreover, it sometimes changes the original ICD code.

   The terms “femoral incomplete fracture” and “femoral shaft fracture” are illustrative examples of the post coordination problem. In fact, “femoral incomplete fracture” shares an ICD code [S72.9] with “femoral fracture”, however “femoral shaft fracture” [S72.3] does not, even though they appear to have the same structure. The difference between them is the dependency inside of the term; “incomplete” modifies “fracture”, whereas “shaft” modifies “femoral”. To address the problem, this paper uses an internal structure of disease names (Figure 1).

This paper presents (1) automatic ICD-coding based on internal structure of disease name and (2) a method to build an internal structure analyzer, including internal structure representation for disease names. The experimental results for internal structure analysis achieved high accuracy (83.7\%), demonstrating the fundamental feasibility of the proposed ICD coding.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{internal_structures.png}
\caption{Internal structures}
\end{figure}
It must be noted that the automatic ICD-coding task in Japanese is more difficult than that in English because Japanese does not use spacing for word/morpheme separation. That difference complicates Japanese processes.

Although experiments described in this paper are related to Japanese medical terms, the proposed method does not depend on a specific language. The proposed method could be applied with the other languages; especially efficient for Chinese, German and so on, which are also without word/morpheme separation.

**Internal Structure Analysis**

The core problems examined in this study are: (1) how to represent internal structure, (2) how to train the analyzer.

1. **Representation of Internal Structure**

An “internal structure of a disease name” is represented as a collection of dependency relations between the morphemes in a disease name. Figure 2 presents an example of “大腿骨転位骨折” which means a femur neck fracture. An arrow indicates a dependency relation. Because of the difficulty in defining the adequate unit (morpheme), we regarded a character as a unit. Japanese character “Kanji” are ideographs.

![Figure 2 - An example of an internal structure](image)

A basic representation of an internal structure is shown above. However, two exceptions exist: omission and contraction.

**Representation for omission**

The first exception is OMISSION as shown below:

\[
\text{心疾患} = \text{心} + \text{疾患}
\]

Therein, “心” is omitted when “心” and “疾患” are compounded into “心疾患” (In fact, “心” itself does not mean “heart organ”). To represent the omission, we introduced a new dependency label “G” and assigned label “D” to the normal dependency relation.

![Representation for omission](image)

In this example, “心” generates a character (actually “臟”) and depends on the generated character, the generated character depends on “患”.

**Representation for contraction**

The framework shown above can deal with another phenomenon: CONTRACTION. Contraction resembles omission, but it differs in that morphemes share the omitted character (“骨” for “大腿骨折”, “膜” for “角結膜炎”).

![Representation for contraction](image)

A dotted square represents a contracted character.

As in the previous example, the dependency label G represents that “臓” “角” generates a character (“骨” “膜”) and depends on the character. The character depends on “折” “炎”.

To process the internal structure adequately, it is necessary to reconstruct omitted and contracted characters. Two points must be specified during reconstruction: 1) the position in which the omission/contraction occurred, and 2) the omitted character. As described in this paper, our internal structure analysis has a framework for the first point.

This study does not examine the inference of omitted characters. That point can be solved using other techniques such as predictive transforms.

**Dependency relation – which is the head?**

Two morphemes in a dependency relation are not symmetrical: one (head) depends on the other (dependent). Here, the question is: Which is which? Regarding the structure of sentences, a head is a constituent that defines the part of speech of a phrase. However, this definition is inapplicable here because a disease name and its constituents are all nouns in Japanese.

We adopted a definition of a head based on an is-a / part-of hierarchy: for a compound noun \( C = \langle \text{unit 1}, \text{unit 2} \rangle \), unit 1 is the head if unit 1 is-a C, else unit 2 is the head. The next step is to apply a part-of relation instead of an is-a relation if both do not exist. For example, in the case of “distal femur”:

- \( \times \) distal femur is-a distal
- \( \times \) distal femur is-a femur

No is-a relation exists. (We adopted a policy that “distal” is a role concept depending on the context. Here, We use a region of “femur”, instead of is-a relation between basic concepts.) Next we present a part-of relation:

- \( \times \) distal femur part-of distal
- \( \circ \) distal femur part-of femur

Therein, “femur” is a hypernym of “distal femur” in the part-of relation, so “femur” is the head and “distal” is dependent.
The internal structure presented above can be analyzed automatically in the framework of dependency analysis studied in the area of natural language processing.

(2) Training of Internal Structure Analyzer

As a training corpus, we annotated the Japanese Standard Disease-Code Master [14]. We randomly chose 114 disease names from C (Neoplasms), 96 from E (Endocrine, nutritional and metabolic diseases), 101 from G (Diseases of the nervous system), 125 from H (Diseases of the eye and adnexa), 123 from K (Diseases of the digestive system), 137 from L (Diseases of the skin and subcutaneous tissue). As an extraction condition, we apply the condition that the length of a disease name is longer than six characters. Then we defined internal structures manually. Shorter words appearing within the target words are also annotated. For example, “肝炎” (two characters) were annotated because it appears within “慢性非活動性肝炎” (eight characters).

As an analyzer, we used MaltParser [12], which is an implementation of shift-reduce parsing. Table 1 presents features used by the analyzer.

**Experiment**

To investigate the performance of the proposed method, we conducted experiments on the performance of internal structure analysis, and discuss its promise for ICD coding. The experimental setting is the following.

**Comparable methods**

PROPOSED: the proposed method described in the previous section.

BASELINE: method by which each character depends on the subsequent character (majority baseline).

**Evaluation Metrics**

According to the sentence parse evaluation manner, we adopted two types of evaluation metrics:

<table>
<thead>
<tr>
<th>Character feature</th>
<th>Dictionary feature</th>
<th>Previous label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Character</td>
<td>4 If a substring of the input disease that end with this character is in the dictionary</td>
<td>11 Dependency labels of Stack[0], Stack[1], Input[0], Input[1]</td>
</tr>
<tr>
<td>2 The type of character</td>
<td>5 Length of the word in (4)</td>
<td>12 Dependency labels of the right/leftmost dependent of Stack[0]</td>
</tr>
<tr>
<td></td>
<td>6 MeSH category of the word in (4)</td>
<td>13 Dependency labels of the leftmost dependent of Input[0]</td>
</tr>
<tr>
<td></td>
<td>7 If a substring of the input disease that end with this character is a suffix</td>
<td>14 Dependency labels of the character left to Stack[0] in the input disease</td>
</tr>
</tbody>
</table>

**Table 2 – Accuracy of Internal Structure Analysis**

*Accuracies with 95 percent confidence intervals. The first line indicates subset of our training corpus: C Neoplasms, E Endocrine, nutritional and metabolic diseases, G Diseases of the nervous system, H Diseases of the eye and adnexa, K Diseases of the digestive system, L Diseases of the skin and subcutaneous tissue. “C-L” is aggregation of 6 categories, i.e. C, E, G, H, K, L. “Overall” is the full set of the
(1) C-ACC: Character level accuracy
(2) W-ACC: Word level accuracy

For the following example (Figure 3), C-ACC is 2/3 and W-ACC is 0/1.

Materials

Using the annotated corpus, the internal structure analyzer was trained and evaluated using five-fold cross validation.

Results

Table 2 shows the obtained result. An example of the system output is portrayed in Figure 4 (dependency labels were all “D” in the example).

The overall accuracies were 95.4% (C-ACC) and 83.7% (W-ACC). The proposed method is superior to the baseline, especially for W-ACC. The E_W-ACC in this table was calculated as C-ACC to the <average length>-th power. It is the expected word level accuracy with an assumption each character’s dependency relations are independent. Actually, W-ACC was superior to E_W-ACC in PROPOSED, although it was not in BASELINE, which shows that the character level-dependency relations are not mutually independent and that the analyzer (proposed method) learned the internal structures well.

The overall accuracy was lower than the reported for an earlier study [11]. As a likely cause, it must be considered that the dependency relation used in our study differs from that in the previous study: the root of the dependency tree was always the end of a word. This difference can influence the result.

Discussion for ICD coding

The proposed internal structure and its analyzer are helpful for precise ICD coding.

Generally speaking, rule-based ICD coding necessitates two points: coding rules and analyzing input diseases. Coding based on the internal structures proceeds as follows: (step 1) obtain the internal structure of the input disease, (step 2) use coding rules by the internal structure acquired at step 1. At step 1, the analyzer described the work explained above. Next, we discuss the coding rules in step 2.

We defined the representation of the coding rule as “an internal structure => an ICD code”. The problem is how to make the rules. The left-hand-side of the rule should be the essence of the concept that the corresponding ICD code (the right-hand-side) denotes.

We propose an example based method: extracting the essential structure from examples sharing the same ICD code.

Figure 5 presents an example of the essential structures of S72.3 and S72.9 based on Figure 1. The structure of S72.3 is the same structure as that of “femoral shaft fracture” in Fig. 1, whereas “incomplete” disappeared in the case of S72.9. That is to say, the essential internal structure for ICD coding is the maximum common subtree of the examples for the ICD code.

The approach described above deals with the order/existence of modifiers (post coordination) that this paper targets. First, the difference of modifiers order is solved by the representation itself because the internal structures of two disease names, which differ from each other merely by the modifier order, are the same. Second, non-essential modifiers disappear when the coding rule is generated.

Once the coding rules have been prepared, the most appropriate rule is sought. An internal structure could fire more than one rule (presuming an input “femoral fracture” and rules presented in Figure 5). Therefore, the question arises: “Which is the correct one?” For this approach, we defined that the rule among candidate rules for which the internal structure (left-hand-side of the rule) size is greatest is the most appropriate.

Coding rules can be adapted even better by changing the comparison strategy: “string match” to “class match”. For example, if “femur fracture” is input, neither of the two rules above fire because of the difference of two strings: “femur” and “femoral”. The solution is “knowledge”, like an ontology, which tells the system that the two strings are of the same class.

As an example, we can presume generation of a rule from two disease names sharing an ICD code, “supracondylar femur fracture” and “distal femoral fracture”. The coding rule generated by the strategy “string match” is the same as S72.9. The information that the left-hand-side contains is insufficient. The strategy “class match” is the solution: “supracondylar” and “distal” belong to the same class, at least in the “femur fracture” context.

The rules can be generated automatically by humans or the internal structure analyzer. Which is the better solution—the analyzer or human—presumably depends on the situation.

![Figure 5 - Examples of coding rules](image)
Previous works

Numerous researchers have examined automatic coding. Such efforts are separable into two groups: (1) term-based [8–10], for which the input is a disease name; and (2) text-based [4–7], for which the input is text such as discharge summaries. The latter have an advantage over the former because they use richer information. Such rich information, however, is not always utilisable. Therefore, it does not motivate our approach.

Another important aspect of coding studies is the coding algorithm such as frequency statistics [10], a Naïve-Bayes classifier [8], and an open registry algorithms [7]. Although various approaches have been proposed, they make use of the input disease name as string, ignoring its internal structure.

Our study differs from these works in the sense that: (1) we use internal structures of disease names, (2) we deal with omission and contraction of characters.

From the perspective of natural language processing, many researchers have examined parsing, i.e., analyzing sentence structures or discourse structures, although they paid less attention to the term structure. Neither is associated with medical informatics. Morphosaurus[13] segments a term into semantically annotated morphemes, however they does not deal with dependency relations.

Conclusion

This paper proposed a new representation for internal structures of disease names. Our character-based representation can deal with peculiar linguistic phenomena— omission and contraction—that previous works have not addressed. For ICD coding, it would be useful to generate coding rules and to analyze the input disease names to be coded.

Acknowledgements

We are deeply grateful to Professor Yuji Matsumoto, Nara Institute of Science and Technology, Japan.

References


Address for correspondence

Emiko Yamada, MMS, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8655, Japan; E-mail: emiko-tky@umin.ac.net