Using N-Gram Method in the Decomposition of Compound Medical Diagnoses

Gergely HÉJA[1], György SURJÁN[2]

[1]Budapest University of Technology and Economics, Department of Measurement and Information System, Magyar Tudósok körútja 2, H-1117, Budapest, Hungary
[2]National Blood Transfusion Service
Karolina út 19-21, H-1113, Budapest, Hungary

Abstract. Compound diagnoses are often assigned to just one disease code. This is a known cause of coding error. Our paper outlines an efficient, cheap and easy to implement method for semi–atutomatic decomposition of such diagnostic expressions. The proposed method is based on n-grams. To verify the method two human encoders were asked to analyse the same set of 92 clinical diagnoses. Their results were compared to the analysis produced by the method. The results demonstrate the reasonability of the approach.

1. Introduction

Indexing of medical diagnoses is a laborious and error-prone task. Providing assistance to the human coding by information systems is an important area of research in medical informatics since many decades [1]. In spite of all the efforts the problem is not yet solved. There are many useful methods, but practically none of them is used outside the local environment where it was introduced.

Either human or computer assisted coding process may use the clinical diagnoses as input information. In both case coding error may arise if a clinical diagnostic statement expresses more than one disease. We call such expressions compound diagnoses. There is a temptation to assign one expression to one code. Human users and computers sometimes do not realise that the correct representation requires more than one code.

In this study we seek for solutions easy to implement, which are able to recognise composite diagnoses and identify their components, the 'atomic' disease concepts. The proposed n-gram based method is statistical and language independent, but not fully automatic. It is not our goal to exclude all human activity from the coding process, but reduce coding errors by effective assistance.

2. Method

In the following:

1. The term ‘Word’ denotes an ‘elementary unit of information’, i.e. in the given approach we are not interested in its internal structure.
2. A sequence of semantically related Words will be called ‘Sentence’
3. By n-gram we will mean n consecutive Words of a particular Sentence.

We capitalise 'Sentence' and 'Word' to discriminate these concepts from word and sentence as grammatical units of language. The grammatical notions of word and sentence are instantiations of our concepts. Since we are dealing with clinical diagnoses the sentence delimitation is not a problem: the diagnosis can be viewed as a Sentence.

Suppose that we have clinical diagnosis expressions from a large number of patient discharge reports. This corpus of Sentences is considered a reference base **R,** which is supposed to be free of errors. That means it consists of sensible, properly spelled diagnoses, all expressing exactly one disease. All n-grams for each Sentence of the sample are then generated (with length of one up to the length of the particular Sentence). In other words, we store all the continuous fragments of each Sentence. A special tag "_FULL_" is added to the n-gram representing the whole Sentence.

Let us suppose now that T is not present in **R** though all of its Words are present in Sentences of **R**. This could happen e.g. when compound expressions or errors are present in T. In such case we can search for the longest set of Words of T which is present together in Sentences of **R**. That means the shortest n-gram of **R** is selected which contains the utmost Words in T. The Word order is disregarded. The remaining set of Words in T can be analysed in the same way until we get an empty set.

Let us suppose that **R** contains only atomic disease concepts and T contains compound ones. Then the found n-grams should denote the atomic concepts in T. (E.g. "diabetes mellitus Hodgkin lymphoma" will be decomposed to "diabetes mellitus" and "Hodgkin lymphoma"). However it could happen that certain Words of a concept in T can extend the n-gram of another one. E.g. in case of "insulin dependent diabetes mellitus non Hodgkin lymphoma" the Word "non" belonging to the concept "non Hodgkin lymphoma" will be assigned to "insulin dependent diabetes mellitus". This faulty decomposition is mainly caused by the method of selecting the longest set of words first.

Therefore we can also use the Word order to augment the precision of the method. It cannot be assumed that T has the same word order as its n-gram in **R** only that it is similar, though it is not likely that Words fall too far away from their original location. The threshold for the allowed distance from the original position can be set empirically. In case of the analysed expressions the value of 2 was optimal.

Now let us take an n-gram which was found in R. It can be checked whether the tag "_FULL_" is assigned to this n-gram. When it is missing it is supposed that the expression represented by the n-gram is incomplete. However when it is assigned, there could be still Words missing from the Sentence, because there could be Sentences in **R** which describe subtypes of the concept described by T. We can also attempt to find the extended n-grams: those n-grams, which contain the Words of the original n-gram. To keep the number of such n-grams low a threshold can be applied. We can define a measure of the co-occurrence likelihood between the original and the extended n-gram: as the occurrence frequency of the extended n-gram divided by the occurrence frequency of the original one. When this measure is too low, then the search in that direction is stopped.

The unknown Words are left out from the search. The missing Words from the extended n-grams can be compared to the unknown Words (we applied statistical similarity measures instead of lexical analysis), therefore typographic errors can be corrected. This measure uses the number of co-occurring characters and their order to compare the two Words.

3. Implementation

It would require too much storage space to store all n-grams independently. It would also take much time to search for other n-grams containing the particular n-gram. Therefore it was decided to store the n-grams in a tree (a compact storage form). E.g. all n-grams beginning with "diabetes" are stored under the node representing "diabetes" (See Figure 1). The number in each box is the occurrence frequency of the n-gram it represents: thus "mellitus - 3" means that the sequence "diabetes mellitus" occurs 3 times.

The tree in Figure 1 stores the Sentences:

1. diabetes mellitus
2. diabetes mellitus type-I
3. diabetes mellitus type-II
4. diabetes insipidus

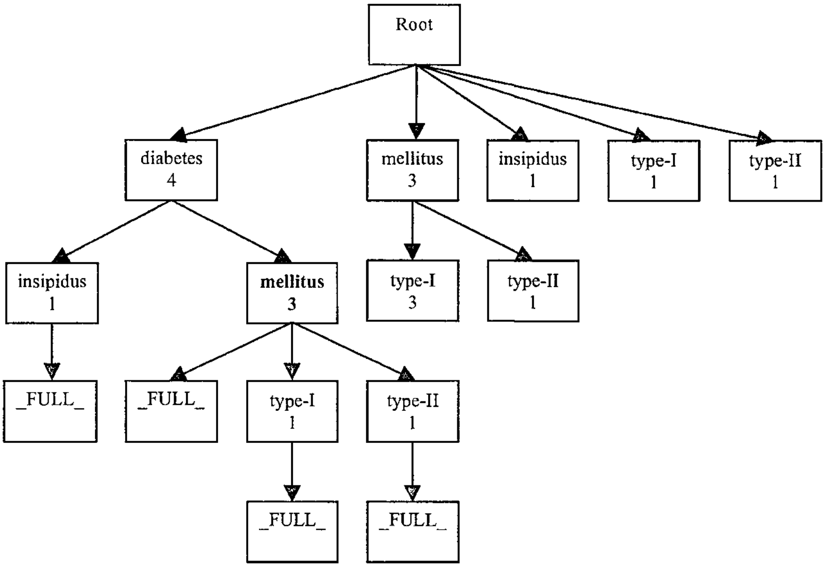

Figure 1

It is easy to search for n-grams which contain the original one at the beginning (e.g. "diabetes mellitus" for "diabetes"). Only the sub-nodes of the node representing the n-gram have to be analysed. To search for n-grams containing the original one at the end (e.g. "diabetes mellitus" for "mellitus") the top-down search practically cannot be used because all Words appear at the root level therefore all nodes in the first layer should be analysed. To be able to search effectively backwards in the tree some modifications are required:

- Every child node has to have a reference to its parent
- The nodes in the nth layer containing the particular Word have to be found easily

The second requirement is needed because it would be very time consuming to inspect all nodes in the given layer of the tree (the number of nodes can be approximated by k^n, where k is the average branching factor and n is the number of words in the given Sentence). To achieve this, nodes containing each particular Word have to be collected. The tree is built up recursively (first the 1-length n-grams are added, then the 2-length, etc.) therefore the ordinal number of the nodes are incremented according to the layers.

The search is done by checking whether the root node is reachable via the Words of the Sentence. First all Words of the Sentence which appear at the n^{th} layer in the tree are selected. Then for all such nodes with bottom-up search it is checked whether the root of the tree is reachable via the other Words of the Sentence.

E.g.: the test Sentence T is "diabetes mellitus"

1. the node "mellitus - 3" is selected for the Word mellitus (because of the backward search we begin at the end of the test Sentence), it is in the second layer of the tree.
2. the parent of "mell. - 3" is "diab. - 3" which denotes "diabetes", T contains this Word, thus the search reached a node whose parent is the root, and the search was successful.

If the search fails, the missing Word(s) are returned. This method is far better for looking for missing Words from the beginning of an n-gram than the top-down search.

4. Experiments

4.1. Gold standard

Diagnostic expressions were taken from a collection of 92 discharge reports. Two independent experienced encoders were asked to decompose the diagnostic expressions when they considered them as compound into terms expressing a single disease [3]. There was not full agreement between the two human encoders in the question, which diagnoses are compound, and which are simple. Out of the 25 diagnoses considered as compound by at least one encoder their judgements were different in seven cases. In three cases encoder *B* considered the expression as a simple disease, while encoder *A* considered them as compound. In further four cases the decomposition was not identical. The results of the two decompositions were reviewed by the authors and we came to consensus with the two encoders. The resulting decomposition is used as the gold standard for the test: from the 92 diagnoses 67 were simple and 25 compound. Altogether 132 expressions were found.

4.2. Evaluation

A corpus of 3081 encoded diagnoses was used as reference base. To perform the necessary calculations a program was written by the authors. The program is able to process the set of Sentences to create the tree, which is stored in a binary file, no general data-base software was used. The 92 diagnoses were decomposed automatically. The result of the analysis of was put into an ASCII text file further analysed with MS Excel.

The performance of the method was compared to the gold standard. To express the correctness of decomposition the terms *precision* and *recall* were used in a slightly modified sense, as it is usual in the document retrieval literature. By *recall* we mean the ratio of the decomposed expression given by the information system and the gold standard to those given by the gold standard. By *precision* we mean the ratio of the decomposed expressions given by the computer system and the gold standard to those given by the computer system only:

$$recall = \frac{humans \,\&\, computer}{humans} \qquad (1)$$

$$precision = \frac{humans \,\&\, computer}{computer} \qquad (2)$$

5. Results

The system decomposed the 92 diagnoses to 293 expressions. From these expressions 109 were found to be valid according to the gold standard. In the case of 21 diagnoses some spelling errors (according to **R**) were found. Our system supports the correction of such errors. After correcting these errors 257 expressions were found, from which 117 were found to be valid against the gold standard. The recall is 88.6 %, the precision is 45.5 %. Altogether 15 expressions were not found by the method. One such expression was faulty identified (the system faulty contracted the fragments of two expressions). 14 expressions contained unknown Words which were essential to the meaning of the expression.

From the 117 valid expressions 62 were fully identified by the system, in 53 cases some minimal manual search was done among the extended n-grams. In 2 cases the hit was only a related concept therefore the search had to be done in the ICD coding system.

6. Discussion

From the 14 expression containing unknown Words 10 referred to medical procedure not to disease. (Physicians sometimes like to refer to history of certain medical procedure in clinical diagnoses. Eg. "Status post appendectomiam" (=Status after appendectomy, Latin) The reference base were created from diagnoses describing diseases therefore it did not contain expressions describing prior medical procedures.

The codes assigned to the diagnoses can also be stored in the tree therefore the method can be used to semi-automatic coding, too.

The n-gram method attempts to integrate the positive features of the traditional vector-space method [4] and of the word-affinity method [5] to assist the decomposition of compound diagnoses. The recall of the method is satisfactory but the precision should be incremented to ease the work of the users. The response time is sufficient to use the method in the daily work.

The presented method was easy to implement and did not required expensive knowledge engineering or language processing work. However it is necessary to create a reference base, consisting of manually coded diagnoses. Since the approach makes it possible to increase incrementally the size of the reference base by a kind of controlled learning, a not too large set is sufficient at the beginning.

References

[1] F. Wingert, Automated indexing based on SNOMED *Meth Inform Med* 24 (1985) 27-34

[2] G. Surján, Question on validity of International Classification of Diseases-coded diagnoses, *International Journal of Medical Informatics* **54** (1999) 77 – 95

[3] G. Surján G and G. Héja, Indexing of Medical Diagnoses by Word Affinity Method, *in proceedings of MEDINFO'2001*, London, United Kingdom, 2-5 September 2001, pp. 276-279

[4] Wiesman F., Hasman H., van den Herik H. J.: Information retrival: an overview of system characteristics, International Journal of Medical Informatics 47 (1997) 5 – 26

[5] G. Surján G and G. Héja, Indexing of Medical Diagnoses by Word Affinity Method, *in proceedings of MEDINFO'2001*, London, United Kingdom, 2-5 September 2001, pp. 276-279