Barriers and challenges of using medical coding systems

Surján, G.

Chapter IV. EFFECT OF CORPUS QUALITY ON PERFORMANCE OF COMPUTER ASSISTED STATISTICAL CODING METHODS

G Surján, G Héja, A Hasman, A Abu-Hanna

Abstract
PURPOSE of this study was to investigate the influence of corpus quality on the performance of corpus-based coding methods.

METHODS Two corpora of manually ICD-coded diagnostic expressions differing in language (German and Hungarian), size (93,863 vs. 53,198 expressions), heterogeneity, and quality were subjected to two different coding methods (based on Naïve Bayes and vector-space). The methods were evaluated using 5 fold cross-validation on both corpora in terms of their ability to place the correct code in a specified rank range. In the German corpus performances were also inspected per ICD chapter. In the Hungarian corpus performances were also evaluated against a silver standard of a dataset of 300 expression-code pairs, where codes emerged by consensus of additional human coders.

RESULTS The corpora differed markedly in terms of redundancy (multiple occurrence of the same expression) and inconsistency (the same expression is assigned to different codes). On the German corpus the vector space method appeared to be slightly superior to Naïve Bayes. The difference was statistically significant but of no apparent practical relevance (52 vs. 50% precision for the first ranked code). On the Hungarian corpus Naïve Bayes drastically outperformed the vector space method (71% vs. 21%). Performance varied strongly per ICD chapter but differences among chapters were higher than differences among the methods. The performance of the Naïve Bayes method improved using the silver standard but this was not the case for the vector space method.

CONCLUSIONS Superiority of one method over another strongly depends on specific quality characteristics of the used corpora. The observed performance discrepancies were attributable to differences in redundancy and inconsistency of the corpora and in performance evaluation one should hence inspect the characteristics of the expression-code relationship. An inconsistent sample, however, might have the advantage of offering different options to a human coder focusing his or her attention to alternative codes that might fit best a given expression.

1. Introduction
ICD coding of clinical cases, often represented as free text in patient discharge summaries, is a compulsory task in many countries. Manual performance of this task is repetitive, costly and error prone. Although the process cannot be fully automated, it can be effectively supported by computers [1]. In the past decades, various coding methods have been proposed in the literature, but none of them is used widely in practice. The general use of such a method requires trust in it. This trust is strongly influenced by the validity of the method, which is determined by evaluating its coding performance.
Computer assisted ICD coding approaches can be grouped into two main types:

A) Approaches that exploit linguistic information beyond the used terms in the discharge summaries (e.g. [2], [3]). For example they may rely on the grammatical structure of sentences and exploit term synonyms. These approaches often rely on natural language processing (NLP) tools.

B) Corpus based approaches using statistical machine learning methods. Instead of relying on external knowledge, such as grammar, these approaches exploit the properties of the joint distribution of ICD codes and the terms found in a corpus. Systems based on these approaches require a sufficiently large corpus of manually coded diagnoses to be input for some learning algorithm but they can be very cost effective and allow for incremental improvement of coding performance [4], [5], [6] and [7]. A special case of corpus based methods is when the terms used in describing the classification itself are used as the training set. Experiments using such a corpus were made by Baud and Lovis et al. [8], [9]. In the Hungarian environment this approach is less promising, since the wording of the Hungarian translation of ICD is markedly different from the daily used medical jargon that appears in the medical record or discharge letter.

These categories of computer assisted coding approaches have been given different names. For example Armakia et al. refer to these categories, respectively, as 'direct' and 'example based' [10] but the various names given to these categories essentially imply the same division lines.

The requirements of the approaches that exploit linguistic information are often not met for many languages especially for languages spoken in smaller communities because the necessary NLP tools are often not developed. For this reason the corpus based approaches provide an attractive alternative and they form the focus in this study.

There is a variety of methods suggested in the literature for implementing corpus based approaches and several papers report on and compare coding performance in terms of measures such as recall and precision (e.g. [11]). In 2007, the Computational Medicine Center's Medical NLP challenge provided a standard set of documents to test different methods on the same corpus to make results comparable. An example of such a comparison on this "standard" corpus is provided by Crammer et al. [12] who evaluated three (combinations of) different methods.

Resnik et al. propose two different kinds of performance measures for computer assisted coding methods [13]. Intrinsic measures compare the result of an automated system to a "gold" standard in terms of precision and recall. Extrinsic methods compare the performance of human users with and without use of computer assisted coding in terms of agreement with a "gold" standard, inter-coder agreement, and intra-coder consistency. To our knowledge none of the previous work addresses the effect of the quality of the used corpora on any of these
performance measures. In this study we quantify the intrinsic performance of different statistical methods using two different corpora that vary in their statistical characteristics. The goal of the study is to investigate whether the performance differences are mainly attributable to the methods or to the differences of the characteristics of the used sample. It was not our aim to compare a large number of statistical methods known from the literature and we rely on relatively simple and easily understood methods. In the following section we describe the used corpora followed by a description of the statistical methods.

2. Material and method

2.1. The corpora and their characteristics

We used two different corpora for our experiments. Both of them consist of diagnostic expressions (noun-phrases) taken from discharge letters. Each diagnostic expression is assigned to ICD by human coders. One is a corpus obtained from two German hospitals (a courtesy of Stefan Schultz, Albert Ludwig University, Freiburg, Germany) containing 93,863 German diagnostic expressions from discharge letters that have been assigned ICD-9 codes. The total word count of the corpus is 267,732, corresponding to an average of 2.9 words per expression. The other corpus is obtained from discharge reports from a surgical department of a university clinic (a courtesy of János Weltner, Semmelweis University Budapest, Hungary) containing 53,198 Hungarian (Latin) diagnostic expressions assigned to the Hungarian adaptation of ICD-10 (WHO International classification of Diseases, version 10). The corpus consists of 138,540 words corresponding to 2.6 words per expression on average. Aside from the language (German and Hungarian) and the specific ICD version used, there were many differences between the two corpora. The German corpus was obtained from several different departments of two hospitals and therefore covers practically all medical domains. In contrast, the Hungarian sample was obtained from just one surgical department, so the covered domain was much narrower focusing on gastrointestinal diseases but without restrictions on co-morbidities.

Both samples are stored in simple text files; each entry contains one clinical diagnostic expression preceded by the ICD code assigned to it by the human coder of the hospital. Figure IV-1 shows examples taken from these corpora where the $ symbol is used as delimiter for the ICD codes for processing purposes.
Chapter IV – Effect of corpus quality

Sample from the German corpus, ICD 9 codes

...$997_9$ Aggregatdysfunktion
$189_0$ Wilms Tumor rechts
$793_8$ Mammographie links, suspekte
$839_7$ Subluxation sepli mit Breitnase
$451_1$ Bein-Beckenvenenthrombose, tiefe
...

Sample from the Hungarian corpus, ICD 10 codes with fifth digit national extension

...$K4490$ hernia diaphragmatica
$I7020$ atherosclerosis universalis
$I8490$ nodi haemorrhoidales
$I10H0$ hypertonia
$C7860$ carcinosis peritonei
...

Examples from the German and Hungarian corpora. Manually assigned codes (ICD-9 for the German and ICD-10 for the Hungarian) are delimited by $ signs and followed by the clinical diagnostic expressions that were coded by human coders.

The following minor pre-processing of typing variations was carried out on both corpora:
- Upper case initials were converted to lower case, but words consisting only of capitals were not converted. So 'Meniere' was converted to 'meniere' but 'TBC' remains in upper case.
- All white-space characters were regarded as word separators.
- A dot following a word was not removed, since the word probably denotes an abbreviation (diagnostic phrases are typically noun phrases and do not form complete sentences with a full stop at the end.)

We did not do any spell-checking.

The two corpora show important statistical differences as shown in Table IV-1. Although the percentage of distinct codes in the two corpora is comparable (3.4% versus 3.8%), the percentage of unique expressions in the German corpus is much higher than that in the Hungarian corpus (about 98% versus 20%). Although the German sample is about 1.8 times the size of the Hungarian sample, the maximum code frequency is higher in the Hungarian sample (3581 times with code “I10.- Hypertension”) than in the German sample (979 times with code “444.2 - arterial embolism or thrombosis of the extremities”). This is mainly due to the higher homogeneity in the Hungarian sample.
Table IV-1
Summary statistics for the German and Hungarian corpora

<table>
<thead>
<tr>
<th></th>
<th>German</th>
<th>Hungarian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of entries in corpus</td>
<td>93,863</td>
<td>53,198</td>
</tr>
<tr>
<td>Distinct codes, N (%)</td>
<td>3,593 (3.8 %)</td>
<td>1,813 (3.4 %)</td>
</tr>
<tr>
<td>Distinct expressions, N (%)</td>
<td>93,624 (99.7 %)</td>
<td>10,689 (20 %)</td>
</tr>
<tr>
<td>Mean code frequency</td>
<td>26.1</td>
<td>29.3</td>
</tr>
<tr>
<td>0/25/50/75/100 percentiles of code frequency</td>
<td>1 / 3 / 8 / 26 / 979</td>
<td>1 / 1 / 3 / 10 / 3,580</td>
</tr>
<tr>
<td>Mean expression frequency</td>
<td>1.002</td>
<td>4.98</td>
</tr>
<tr>
<td>0/25/50/75/100 percentiles of expression frequency</td>
<td>1 / 1 / 1 / 3</td>
<td>1 / 1 / 1 / 2 / 3,635</td>
</tr>
<tr>
<td>Percentage of codes that appear only once</td>
<td>14.8% (532/3593)</td>
<td>32.7% (594/1813)</td>
</tr>
<tr>
<td>Most frequent code</td>
<td>444.2 (arterial embolism or thrombosis of the extremities)</td>
<td>110.- Hypertension</td>
</tr>
</tbody>
</table>

The Hungarian sample showed a high degree of coding inconsistencies: the same expression was often assigned to many different codes. The five most inconsistently coded expressions are diabetes mellitus (31 different codes), ISZB (Hungarian acronym for ischemic heart disease, 21 different codes), hepatic cirrhosis (16 different codes), anaemia (16 different codes), and coronariasclerosis (sclerosis of the coronary arteries – 15 different codes). This is an intriguing property of a corpus’ quality, and we will investigate its effects on coding performance.

2.2. The statistical methods
Two statistical methods, vector space and Naïve Bayes, were applied to both corpora. The analysis on both of these corpora was based on 5-fold cross validation. This means that each corpus was randomly divided into 5 equally sized and disjoint parts. Each part was used once as test sample while the rest of the corpus was used as the training sample.

2.2.1. The vector space method
Vector-space methods are based on the so called word-document matrix that represents a certain corpus of text by a matrix having rows corresponding to the words appearing in the corpus and columns corresponding to the “documents” that form the corpus. In our case a document corresponds to the diagnostic expression. In the training set this document is also associated with a given code. This matrix can be either binary or weighted. Binary matrices have 0-s or 1-s as cell
values indicating whether the corresponding word appears (1) or not (0) in the
Corresponding document. Weighted matrices might use local and global weights. A
Local weight reflects the idea that words appearing more frequently in a document are
given a greater local weight than words that appear less frequently. This weight
improves the recall of relevant documents and can be quantified by the word
frequency (also referred to as term frequency, TF). Global weighting reflects the
idea that words that appear in a small number of documents tend to be more
significant than those distributed more uniformly over the documents. This weight
improves the precision of the retrieved documents. For a given word this weight
may be quantified by the inverse of the frequency of the documents that contain
this word. Commonly one uses the logarithm of this inverse frequency.

These two types of weights are combined by the so called TF×IDF approach [14]
where TF indicated the term (word) frequency and IDF the inverse document
frequency. There are different schemes to specify TF and IDF but a common
approach is to assign the following quantity to the cells of the word-document
matrix:

$$TFIDF_{i,j} = O_{i,j} \times \log(N / n_j)$$

where TF is quantified by $O_{ij}$ which is the number of occurrences of the $j$-th word
in the $i$-th document, $N$ is the total number of documents in the corpus, and $n_j$ is
the number of documents that contain (at least once) the $j$-th word. The IDF is
quantified by $\log(N / n_j)$. Each column (document) in the matrix can be represented
as a vector in a multi-dimensional word space (consisting of all words). Then one
may define similarity measures between these vectors (documents).

A common similarity measure is defined by the (cosine of the) angle of the two
document vectors in the word space. This approach is used to find similar
documents in document retrieval problems [14], [15] and [16].
The cosine similarity of two documents is the inner product (also known as the
scalar or dot product) of the two corresponding document vectors, in the word
space, divided by the product of their Euclidean norms. This division normalizes
the inner product and neutralizes the dominance of longer documents with more
words. In our experiments we used the vector space method with

$$TFIDF_{i,j} = O_{i,j} \times \log(N / n_j)$$

and with the cosine similarity measure.

In the context of computer assisted coding the corpus consists of diagnostic
expressions (documents) that are associated with a code, while a query is an
expression for which we are seeking relevant expressions and more importantly,
their corresponding codes. If there are expressions with a sufficient level of
similarity in the sample, the codes associated with those expressions are retrieved
in the belief that their codes include one that might be appropriate for the query
expression as well. To resolve the zero frequency problem where a query word
does not appear at all in a document we resort to pseudo-counts based on Laplace
smoothing. We specifically chose for the variant in which 1 is added to the
Chapter IV – Effect of corpus quality

numerator and 2 to the denominator when estimating the relative frequency of a word.

2.2.2. The Naïve Bayes method

One may treat the coding problem as assigning a probability to each candidate code based on the query words \( w_1 \) to \( w_n \) and then ranking the codes based on these probabilities. Formally we are seeking \( P(\text{code}_i \mid w_1, \ldots, w_n) \) which under the Naïve Bayes assumption of class conditional independence, is equal to \( P(w_1 \mid \text{code}_i) \cdots P(w_n \mid \text{code}_i)P(\text{code}_i)/P(w_1, \ldots, w_n) \). The denominator is a constant that will not affect the ranking results and is not calculated.

The class conditional independence assumption is often not valid, but it is practically useful because we are interested in ranking codes and not in the absolute probabilities, which might be inexact [17]. Again to resolve the zero frequency problem when estimating the relative frequency of a word we resort to pseudo-counts based on Laplace smoothing as in the vector space method.

2.2.3. String matching control

Whenever the training sample contains a perfect match to the query diagnostic expression, the use of the statistical methods described above would not be necessary, provided that the samples are fully consistent in the sense that an expression is unambiguously assigned to a code. But since consistency is hardly met with manually coded data, finding a perfect string match is not enough to find the right code. In principle all statistical methods can be used to detect inconsistent cases and can be used to clean up the learning sample, as was described [18]. However, this method necessitates a large human effort and is outside the scope of this study. Instead, the number of cases for which there was a perfect string match has been recorded in order to see how much one would gain from using the statistical methods.

2.3. Evaluation approach and performance measures

In cross-validation the 5-folds were identical for the methods considered. For each method the 10 top ranked codes were recorded. We take note of the rank of the true code if it was included in the top 10 ranks. If it was not among the top 10 ranks, then we note whether it appeared in the training set (of that fold) or not. Finally, we record the status of perfect string matches: 0: when one perfect match of the query expression was found but had a different code than the query, 1: when one perfect match was found with the same code, 2: when multiple string matches were found, and all codes are the same, 3: when multiple string matches were found, where some codes are the same but others are different, 4: when multiple string matches were found and all codes are different, and 5: when no string match was found).
We calculated the following performance measures:

- the percentage of the query diagnostic expressions that were assigned the correct code in the top 10 ranks and the rank of that code.
- the percentage of query expressions for which the correct code could not be retrieved at all although it was in the training fold.
- the percentage of query expressions for which the correct code was missing from the learning sample.

These figures were considered in the context of the number of expressions for which there was a perfect string match for the query expression.

We performed the same evaluation on both corpora. In order to separate the effect of the different methods and the effect of the different corpora, the German test set samples were inspected per chapter of ICD-9.

In real world corpora there is a considerable amount of noise where expressions are inconsistently coded by human coders. Hence there is no gold standard of ICD coding [19] and the learning algorithms will tend to reproduce the erroneous human coding. We resorted to a silver standard where 300 diagnostic expressions from each of two Hungarian test samples were randomly selected. Those terms for which the codes were different were sent to a third independent expert at another institute and were recoded again. This third coder was blind to the codes proposed by the others. When at least two codes were identical then this common code was chosen, when all three were different, the first two experts discussed the case and achieved a consensus. These silver standard sets then were tested again using the same learning set.

### 2.4. Computing environment (SW HW)

The experiments have been carried out with the help of a stand alone application developed by one of the authors (G.H.) written in visual C++. This software was run on an Intel dual core Pentium 4 PC, 3.20 GHz processor with 1 GB RAM. In this environment the learning phase takes about 15 minutes or less, the testing phase requires about 1 hour for the Hungarian and 3 hours for the German corpus (note the difference in their sizes).

### 3. Results

Table 2 shows the results for the Naïve Bayes and the vector space methods for the German corpus. A cell in the table shows the two method’s performance in terms of the 5 fold average percentage (and average number) of query expressions for which the correct code was found in the indicated rank range. For example, the Naïve Bayes method’s top three ranks included the correct code in 63% of the test query expressions (corresponding to 11867.8 cases on average), while this was 67% (12475.5 cases) for the vector space method. For each rank range the vector space’s performance was statistically significantly better than the Naïve Bayes method (Paired t-test p < 0.05). This statistical significance does not however imply practical relevancy as the differences are quite small.
Table 3 shows the results for the Naïve Bayes and the vector space methods for the Hungarian corpus. Here a drastically different picture emerges. The Naïve Bayes method outperformed the vector space method in all rank ranges; not only in terms of statistical significance but the differences have a clear practical relevance.

Table IV-2
Results on the German sample (n=18775). A cell in the table provides the 5 fold average percentage (and average number) of expressions for which the correct code was found in the indicated rank range. For example in the Naïve Bayes method the correct code was found in the first 3 ranks in 63%, on average, of the test query expressions (corresponding to 11867.8 cases). Numbers printed in bold face indicate better performance.

<table>
<thead>
<tr>
<th>Rank range</th>
<th>Naïve Bayes % (N)</th>
<th>Vector space % (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>50 (9405.0)</td>
<td>52 (9810.5)</td>
</tr>
<tr>
<td>1-2</td>
<td>59 (11148.2)</td>
<td>63 (11739.3)</td>
</tr>
<tr>
<td>1-3</td>
<td>63 (11867.8)</td>
<td>67 (12475.5)</td>
</tr>
<tr>
<td>1-4</td>
<td>65 (12284.0)</td>
<td>69 (12868.3)</td>
</tr>
<tr>
<td>1-5</td>
<td>67 (12534.2)</td>
<td>70 (13128.8)</td>
</tr>
<tr>
<td>1-10</td>
<td>70 (13178.8)</td>
<td>73 (13780.3)</td>
</tr>
</tbody>
</table>

Table IV-3
Average results on the Hungarian sample, two methods compared (n=10641).

<table>
<thead>
<tr>
<th>Rank range</th>
<th>Naïve Bayes % (N)</th>
<th>Vector space % (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>71 (7571.4)</td>
<td>21 (2184.8)</td>
</tr>
<tr>
<td>1-2</td>
<td>83 (8802.4)</td>
<td>34 (3648.8)</td>
</tr>
<tr>
<td>1-3</td>
<td>87 (9245.8)</td>
<td>45 (4768.2)</td>
</tr>
<tr>
<td>1-4</td>
<td>89 (9474.0)</td>
<td>53 (5623.0)</td>
</tr>
<tr>
<td>1-5</td>
<td>90 (9614.0)</td>
<td>60 (6338.6)</td>
</tr>
<tr>
<td>1-10</td>
<td>93 (9878.2)</td>
<td>76 (8123.2)</td>
</tr>
</tbody>
</table>

Figure IV-2 shows the performance of the two methods per ICD-9 chapter. The graph shows the fivefold average precision (the percentage of cases in which the
first ranked code was identical to the originally manually assigned one) for each chapter of ICD-9. It is apparent that the results vary strongly from chapter to chapter and that this is independent of the method. With the exception of Chapter 16, the vector-space method performs slightly better than the Naïve Bayes, but the differences again have no practical relevance.

![Graph showing performance of two methods in the 18 ICD-9 chapters](image)

**Figure IV-2 Performance (percentage of times in which the observed code was ranked in the first place) of the two methods in the 18 ICD-9 chapters in the German sample.**

Table IV-4 shows the performance results with the silver standard on the Hungarian sample. In general, the Naïve Bayes’s performance slightly improved, especially in respect to the first ranked code (78% vs. 71%). This did not hold for the conventional vector space method.
Table IV-4
Average results of the silver standard using the Hungarian training samples (n=300).

<table>
<thead>
<tr>
<th>Rank range</th>
<th>Naïve Bayes % (N)</th>
<th>Vector space % (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>78 (232)</td>
<td>19 (56)</td>
</tr>
<tr>
<td>1-2</td>
<td>85 (253)</td>
<td>33 (98.5)</td>
</tr>
<tr>
<td>1 – 3</td>
<td>88 (262.5)</td>
<td>45 (134.5)</td>
</tr>
<tr>
<td>1 – 4</td>
<td>90 (268)</td>
<td>53 (160)</td>
</tr>
<tr>
<td>1 – 5</td>
<td>90 (269.5)</td>
<td>61 (181.5)</td>
</tr>
<tr>
<td>1 -10</td>
<td>92 (275.5)</td>
<td>76 (228)</td>
</tr>
</tbody>
</table>

4. Discussion and Conclusion

Much experimental comparative work on coding methods concludes with statements about the superiority of one method over another. Similarly, based on the German corpus, we could conclude that the performance of the vector space method was superior to the Naïve Bayes method. By the same token, if we performed our experiments only on the Hungarian corpus we would conclude the opposite: the Naïve Bayes would be clearly superior. However the main goal of this paper is to show that statements about superiority of methods were dependent on specific qualities of the corpora used and expose these qualities.

Aside from language, which might play some modest role in explaining the differences, one needs to inspect the expression-code relationship in each corpus to understand the discrepancy between the dramatically different results in the two corpora. In the German corpus a diagnostic expression rarely occurs more than once. Although one code is expressed on average in 26 different ways, an expression does not imply much ambiguity in terms of the correct code and both methods perform relatively well. In contrast, in the Hungarian corpus the same expression is assigned to many codes. In fact more than 80 percent of all test expressions have a perfect string match at least once, but often much more frequently, with expressions in the learning sample in each cross validation fold. In these cases the algorithm would perform well if it provides a high ranking to the most frequent code associated with these perfectly matching expressions. The vector space method, however, often fails to find this code. To understand why, consider the following scenario. Assume that a query expression $q_e$ consists of the words $w_1$ and $w_2$ and is labelled with code $c_1$. Now assume that $q_e$ appears 10 times in the training set, 5 times with code $c_1$, 3 times with code $c_2$, 1 time with $c_3$, and 1 time with $c_4$. First, note that the words $w_1$ and $w_2$ are associated with at least 4 codes. This means that the inverse document frequency (IDF) is lower for these words than in the German corpus because more “documents” (codes) now include these words. Second, the cosine similarity measure, which considers the angle between vectors in the word space, does not take into account the frequency with
which \( w_1 \) and \( w_2 \) appear with the correct code, \( \text{code}_1 \). Hence the vector space method does not have reason to prefer \( \text{code}_1 \) above the other codes \( \text{code}_3 \), \( \text{code}_4 \), and \( \text{code}_5 \). In conclusion, these latter codes will often prevent a high ranking of the most frequent code associated with the query expression. Naïve Bayes, in contrast, adequately considers the conditional frequencies of the words given the code, \( P(w_1 \mid \text{code}_i) \) and \( P(w_2 \mid \text{code}_i) \) for each \( \text{code}_i \), in addition to the a priori probability of \( P(\text{code}_i) \).

The performance based on the silver standard confirms these observations. The labels of the test set are now based on expert opinion. This opinion would tend to concord with the majority code assigned to an expression. In the scenario above this means that the query expression \( q_e \) will most of the times be labelled by the experts with \( \text{code}_1 \) which is the most common code that is assigned to \( q_e \), even if its original coding was different. This means that the Naïve Bayes would more often find the correct code with expert coding because in the cases in which \( q_e \) was actually labelled as \( \text{code}_2 \), \( \text{code}_3 \), or \( \text{code}_4 \) it most probably predicted it as \( \text{code}_1 \) which is the expert opinion’s choice. Our experience is hence in line with the assumption of Pakhomov et al, who postulate that for a given diagnosis correct codes appear more often in large corpora than wrong ones [17].

The results in the German corpus by ICD-9 chapter revealed another lesson about the relationship between methods and corpus. There were clear differences between the performances in the various chapters and in general the vector space method was better in the great majority of the chapters. However, the changes of performance within chapters were much more significant than discrepancies in performance between the methods. Hence, the performance of various statistical methods might depend, as in our experiments, more on the quality and characteristics of the used samples than on the features of the method per se.

One implication of our work is that one has to be very cautious when comparing different publications concerning their reported corpus-based performances. If the results of different publications are based on the use of different corpora, the differences may be due to the different corpora rather than to intrinsic features of the different methods. An important issue to scrutinize is the quality of the relationship between expressions and codes.

An important observation that can be made is that using an inconsistent training sample might actually have some benefits above a completely consistent sample in which an expression is always assigned to the same code. Note that coding algorithms are essentially trying to reproduce human coding, including any errors that humans make. This has however the benefit when a coder is offered a (ranked) set of possible candidates providing different “shades” of meaning. Some option codes can draw the attention of the coder to the various possibilities and help him or her to be exact about what they would like to code. To illustrate suppose a patient record includes the following two diagnostic expressions:

\begin{itemize}
  \item \text{Insulin dependent diabetes mellitus,}
  \item \text{Diabetic retinopathy.}
\end{itemize}
For the first expression the "theoretically" correct code is E10.9, which is expressed as "insulin dependent diabetes mellitus without complications." From the second expression it is also clear that this patient has ophthalmic complications, and hence the right code for this condition is E10.3 (Type 1 diabetes mellitus with ophthalmic complications). Our Hungarian sample contained 35 different codes for diabetes mellitus, among which both E10.9 and E10.3. A system that offers different options (E10.9, E10.3 etc) to the user will tend to improve ICD coding as long as wrong options, under any circumstance, are not highly ranked. To illustrate, the code E14.9 would be a good approximation for the expression diabetes mellitus for 1009 expressions, while the next frequent code (E14.7) appeared, incorrectly, in only 86 times.

In conclusion, comparative experimental research of corpus-based coding should consider the quality of the corpora used, especially in terms of the consistency of assigning codes to expressions. When considering the ICD family of codes, performance differences within chapters may be more relevant than discrepancies in performance between methods. Finally, slight coding inconsistencies in corpora may actually help the incremental coding effort by drawing the attention of the human coder to various options.

Acknowledgements
We thank Stefan Schultz and János Weltner for providing the German and Hungarian corpora.

References
Chapter IV – Effect of corpus quality


