Research Paper ■

Automating the Assignment of Diagnosis Codes to Patient Encounters Using Example-based and Machine Learning Techniques

SERGUEI V.S. PAKHOMOV, PHD, JAMES D. BUNTROCK, MS, CHRISTOPHER G. CHUTE, MD, DRPH

Abstract **Objective:** Human classification of diagnoses is a labor intensive process that consumes significant resources. Most medical practices use specially trained medical coders to categorize diagnoses for billing and research purposes.

Methods: We have developed an automated coding system designed to assign codes to clinical diagnoses. The system uses the notion of certainty to recommend subsequent processing. Codes with the highest certainty are generated by matching the diagnostic text to frequent examples in a database of 22 million manually coded entries. These code assignments are not subject to subsequent manual review. Codes at a lower certainty level are assigned by matching to previously infrequently coded examples. The least certain codes are generated by a naïve Bayes classifier. The latter two types of codes are subsequently manually reviewed.

Measurements: Standard information retrieval accuracy measurements of precision, recall and f-measure were used. Micro- and macro-averaged results were computed.

Results: At least 48% of all EMR problem list entries at the Mayo Clinic can be automatically classified with macro-averaged 98.0% precision, 98.3% recall and an f-score of 98.2%. An additional 34% of the entries are classified with macro-averaged 90.1% precision, 95.6% recall and 93.1% f-score. The remaining 18% of the entries are classified with macro-averaged 58.5%.

Conclusion: Over two thirds of all diagnoses are coded automatically with high accuracy. The system has been successfully implemented at the Mayo Clinic, which resulted in a reduction of staff engaged in manual coding from thirty-four coders to seven verifiers.

■ **J Am Med Inform Assoc.** 2006;13:516–525. DOI 10.1197/jamia.M2077.

Introduction

The system described in this article is designed to assign classification codes from a pre-defined classification scheme to the diagnoses generated at the Mayo Clinic and entered into the patients' problem list in the form of natural language statements. These classification codes are subsequently used in clinical research and are a part of the records-linkage system created under the auspices of the Rochester Epidemiology Project.[1, 2] Currently, the task of assigning classification categories to the diagnoses is carried out manually by the Medical Index staff. The volume of medical records generated at the Mayo Clinic is overwhelming the manual classification capacity resulting in a significant backlog. The purpose of the Automatic Categorization System (Autocoder) presented in this article is to improve the coding staff's efficiency by partially automating the coding process with a computer system trained on the rich history of coding experience generated by the Medical Index over the last ten years.

Affiliation of the authors: Division of Biomedical Informatics, Department of Health Sciences Research, Mayo Clinic, Rochester, MN.

The authors thank Dr. Robyn McClelland for her assistance with designing the test sets and providing us with excellent statistician's expertise and perspective. The authors also thank Barbara Abbot and Deborah Albrecht for helping us with developing the reference standard test set and for sharing their expertise and experience in medical coding.

Correspondence and reprints: Serguei V.S. Pakhomov, PhD, 200 First Street SW, Rochester, MN 55905; e-mail: <pakhomov.serguei@mayo.edu>.

Received for review: 02/06/06; accepted for publication: 05/30/06.

Background

Medical Index and Electronic Medical Record

The Medical Index continually updates a set of databases that serve as an index to the patient's electronic medical record (EMR) at the Mayo Clinic. The EMR is generated for each patient and consists of a set of clinical notes, lab test results, prescription orders, demographic information, and problem list statements. The Medical Index database is an institutional resource used to identify patient records for epidemiological research, statistical analysis, administrative reporting, and quality control. The index is created by coding and classification of diagnoses from a patient's EMR using a physician generated patient problem list as the primary source of information.

The problem list consists of diagnostic statements dictated by physicians and transcribed into clinical notes as part of regular documentation for each patient visit. Mayo Clinic clinical notes are structured using the HL7 Clinical Docu-

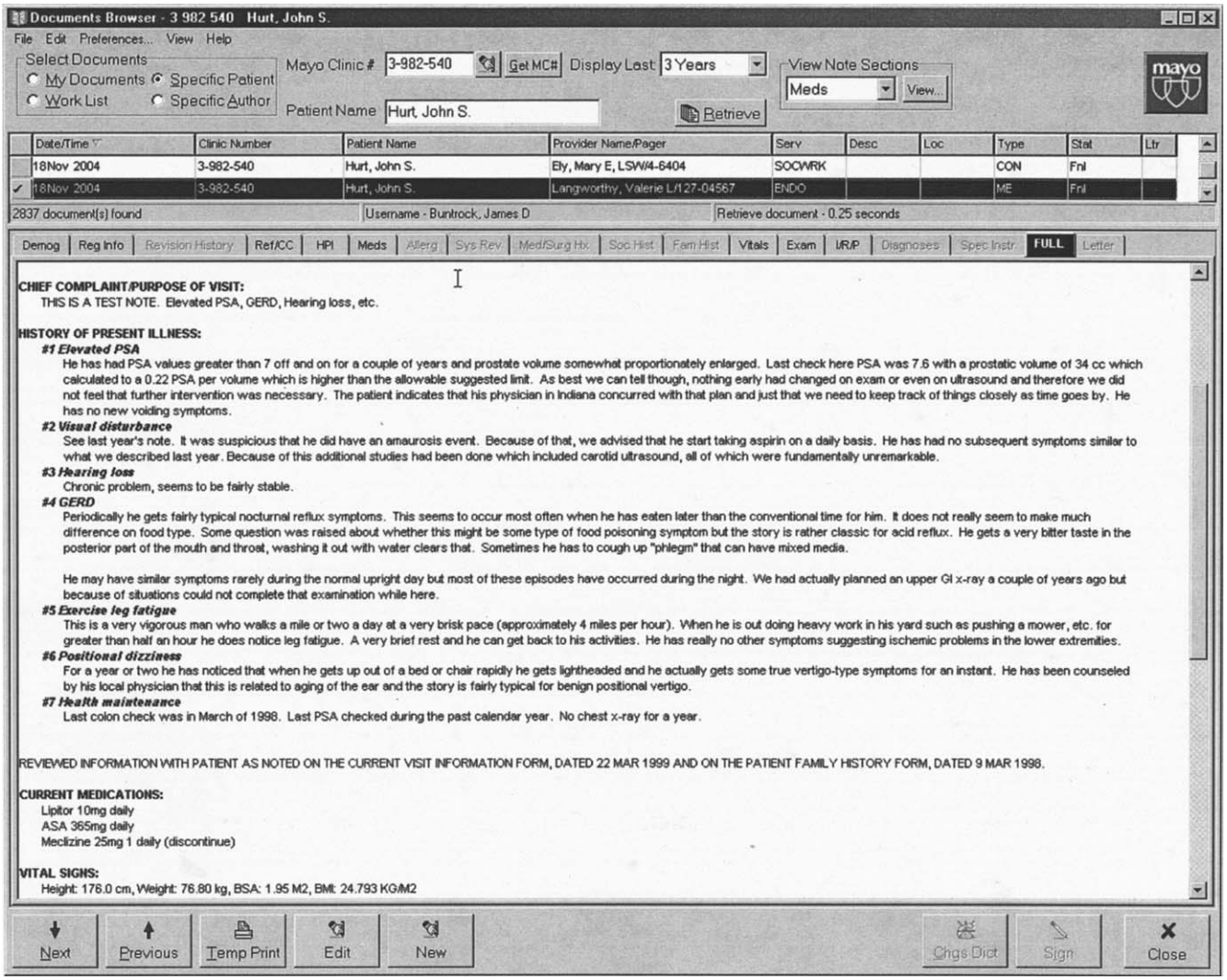

Figure 1. An illustration of a clinical note for a non-existent patient.

ment Architecture (CDA) specification consisting of sections including chief complaint, history of present illness, impression/report/plan, and final diagnosis. An illustration of a typical clinical note for a non-existent patient is provided in Figure 1. The numbered items under the history of present illness section illustrate the kinds of diagnostic statements that are subsequently coded and entered into the Medical Index database.

HICDA Classification

Hospital International Classification of Disease Adaptation[3] (HICDA) is an adaptation of ICD-8[1] (International Classification of Diseases) for hospital morbidity (a.k.a. HICDA-2). While ICD-8 is outdated, its Mayo Clinic adaptation continues to be used internally to maintain the continuity of the Medical Index and the Rochester Epidemiology Project for on-going longitudinal studies. The system described in this article has been trained on examples coded in HICDA; however, our experience and insights into the system architecture and process are generalizable to other hierarchical classification schemes.

HICDA is a hierarchical classification with 19 root nodes and 4,334 leaf nodes. Since 1975, it has been expanded at the Mayo Clinic to comprise 35,676 rubrics or leaf nodes. Each leaf node is assigned an eight digit code with the following internal structure: 12345 67 8. The first five digits constitute the first level below the roots and derive from the original HICDA-2. The seven and eight digit codes constitute Mayo extensions and form the present leaf level. The numeric value of the first five digits determines which root the category reflected by the five digits belongs to. For example, the concept of ANEMIA has the code 02859 21 0 where the first five digits comprise a general category consisting of the following concepts: ANEMIA, NORMOCHROMIC ANEMIA, SECONDARY ANEMIA, IDIOPATHIC ANEMIA, HYDREMIA, OLIGOCYTHEMIA and RUNNER'S ANEMIA. Consistent with the historical chaptering structure of the ICDs, the fact that the numeric

[1]ICD-8 is the 8th edition of the International Classification of Diseases. ICD-10 is the most current edition and is used for mortality coding world-wide; ICD-9CM (Clinically Modified) is usually used for billing in the United States. The Mayo research coding system is based upon a morbidity oriented adaptation of ICD-8, HICDA-2 which has been augmented with concepts whose granularity and relevance are more appropriate for health science research.

value of the first five digits lies in the range between 2800 and 2899 indicates that this set of concepts belongs to an even broader category of "Diseases of Blood and Blood-forming Organs." The top 19 nodes include such categories as "Infective and Parasitic Diseases," "Diseases of Blood and Blood-forming Organs" or "Neoplasms."

Previous Work on Automatic Categorization of Medical Text

As with most automatic text classification problems, the goal is to map between unstructured or semi-structured text and a set of predefined classification categories. The mapping can be either manually constructed, by using expert knowledge transformed into a set of rule-based classifiers, or it can be learned automatically from a set of previously manually categorized training samples. For the latter approach machine learning techniques usually excel with larger training sets.

The classification problems that have been investigated in the past are just as varied as the machine learning algorithms that have been used to solve these problems. Linear Least Squares Fit,[4] support vector machines,[5] decision trees,[6] Bayesian learning,[7] symbolic rule induction,[8] and maximum entropy[9] are just a few algorithms that have been applied to classifying e-mail, Web pages, newswire articles, and medical reports. Much of the previous work in biomedicine has focused on classifying narrative clinical reports to identify patients with a specific condition or a set of conditions.[6, 10-15]

The work reported in this article focuses on classifying only the physicians' diagnoses made during out-patient visits. This problem has also been previously investigated in other settings.[16] Gundersen et al.[17] present a system designed to assign diagnostic ICD-9 codes to the free text of admission diagnoses. This system encodes the diagnoses using categories from a standard classification scheme based on a text parsing technique informed with semantic information derived from a Bayesian network. Another system similar to ours has been developed by Yang et al.[18] This system (ExpNet) comprises a machine learning method for automatic coding of medical diagnoses at the Mayo Clinic. The ExpNet technique offered improvement in scalability and computational training efficiency over previous techniques (Linear Least Squares Fit and Latent Semantic Indexing) with an average precision of 83% and recall of 81% on diagnostic text.[19] This automatic coding method worked well on smaller phrases with less than five or six words and a single diagnostic rubric, but performed poorly on larger phrases with multiple diagnostic rubrics. We are building on the groundwork laid by Yang and Chute[20] by scaling it up with a hybrid approach consisting of example-based classification and a simple but robust classification algorithm (naïve Bayes) in order to improve the efficiency of diagnostic coding.

Methods

The general architecture of the Autocoder and the workflow are illustrated in Figure 2. The diagnostic problem list statements (numbered items in Figure 1) are assigned HICDA diagnostic codes. A portion of these are then stored directly in the Medical Index database and the remaining are post-processed by human coders. Whether a classification is entered into the database without manual review depends on the Autocoder's level of confidence described in detail in the subsequent sections.

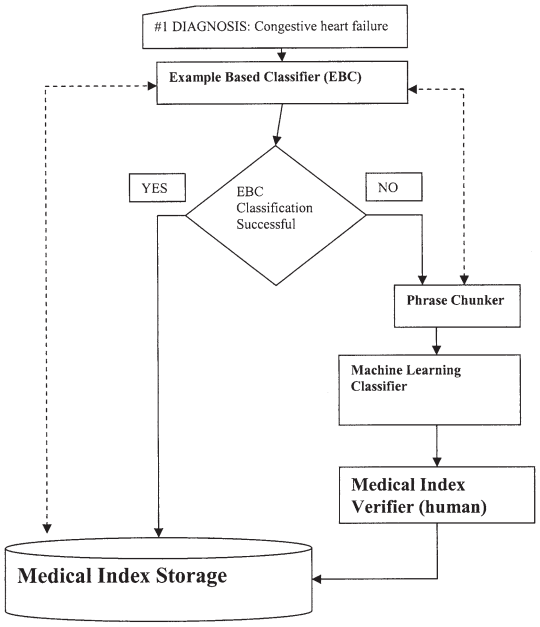

Figure 2. Automatic Medical Index Classification Architecture.

Design Objectives

The main objective of this system is to make the manual coding process more efficient and thus increase its throughput without any significant loss in accuracy. The system is designed to increase the throughput in two ways: by generating classification suggestions for further review and by generating final classification decisions that will not be manually reviewed. In order to make an automatic classification decision without subsequent manual review, a high level of accuracy is required. Our objective was to maximize the accuracy at a level exceeding 95% for both precision and recall. The requirements for making classification suggestions were less stringent because of the subsequent manual verification. Lower accuracy would create a negative effect on the manual verification efficiency but would not affect the overall quality of the resulting index. Nevertheless, our objective was to maximize the accuracy of this part of the system as well. Another important design objective was to modify the coding application to accommodate the new requirements and to maximize the speed at which the Medical Index staff would be able to verify automatically assigned codes.

Autocoder Architecture

The current implementation of the Autocoder is based on two main techniques: example-based classification and machine learning classification (Figure 2). A diagnostic statement derived from the problem list of a clinical note (e.g., "#1 Congestive heart failure") is presented to the example-based classification component. If this component is successful in finding the correct classification code, then the diagnostic statement is entered directly into the Medical Index database without manual review. If the example-based classification fails to identify the code(s) for the diagnostic statement, then the statement is routed to the machine learning component, which produces a set of suggestions ranked by the confidence of the machine learning component. These suggestions are then verified by the Medical Index staff and are subsequently entered into the Medical Index database.

Example-Based Classification Component

Example-based classification leverages the repository of diagnostic statements that have already been manually classified by the Medical Index staff at the Mayo Clinic. The main assumption behind example-based classification is that diagnostic statements are highly repetitive and that, in the absence of major changes in the classification scheme, new diagnoses can be accurately coded automatically by simply looking them up in the database of previously classified entries. For example, if "hypertension" has been coded with code A 1,000 times and code B ten times, we can assume that code A is correct and code B is incorrect. However, this technique raises a number of technical challenges. One such challenge is the noise in the data that results from occasional misclassification or disagreement between Medical Index staff. The disagreement may be precipitated by various levels of experience or by the vagueness of the classification distinctions in HICDA. Part of the example-based classification is to filter out unlikely classifications for a given diagnostic statement coupled with the gender of the patient based on the frequency of their co-occurrence.

Diagnostic entries Gender: Gender is an important predictive feature that is to be taken into account during classification into HICDA because some of the HICDA categories are sensitive to gender distinctions. For example, "pelvic abscess" is coded as 06821140 if the patient is male and 06169111 if the patient is female. Thus, for convenience of exposition, we introduce the notion of a "diagnostic entry" which consists of three components: the diagnostic statement, patient's gender, and the HICDA classification code that was manually assigned.

Multiple codes: Example-based classification relies on the relative frequency of diagnostic entries. All diagnostic entries are considered likely candidates and are arranged in a simple database table. Multiple HICDA codes for a diagnostic entry are combined into a single compound code. For example, "Acute bronchitis, hypertension" is a diagnostic statement that has been assigned two classification codes: 04890112 (BRONCHITIS, ACUTE) and 04010210 (HYPERTENSION, NOS—HPT). In order to maximize the precision of the example-based classification technique we treat such multiple coding as a single multi-code category: 04890112_04010210. The motivation for this approach and its validation have been reported elsewhere.[21]

Frequency filtering The following example illustrates the methodology for filtering likely candidate classifications. One of the most common diagnoses at the Mayo Clinic is "Hypertension." In our data, the diagnostic statement that consists only of the string "Hypertension" appears 168,999 times. This string happens to be coded in 176 different ways forming many different diagnostic entries; however, only two of these entries appear with high frequency:

1. Hypertension—female—04010210 (hypertension) occurs 89,507 times
2. Hypertension—male—04010210 (hypertension) occurs 79,269 times

Notably, the two next most frequent entries occur only five times:

3. Hypertension—female—02500110 (diabetes mellitus) occurs five times
4. Hypertension—male—02500110 (diabetes mellitus) occurs five times

These low frequency entries constitute errors and need to be filtered out as noise. In order to classify a newly encountered diagnosis of "Hypertension" for a male patient, we query the database to find all diagnostic entries where the diagnostic statement matches "Hypertension" exactly and gender component is "male" (Example 2 and 4) and then sort the result set based on the frequency of the diagnostic entries. We then apply two threshold parameters:

A. MIN_EVENT_FREQ—minimum event frequency

B. MAX_NUM_CAT—maximum number of top categories

The first parameter, MIN_EVENT_FREQ, is the minimum threshold related to the diagnostic event frequency. If the frequency falls below this threshold, the HICDA categories associated with the diagnostic event are considered correct. The second parameter, MAX_NUM_CAT, controls how many of the top most frequent events the system has to consider as potential candidates for category assignment. We found the optimal set of parameters to be 25 for MIN_EVENT_FREQ and 2 for MAX_NUM_CAT as detailed in the Status Report section.

In the example with a male patient whose record contains a diagnostic statement of "Hypertension," the MAX_NUM_CAT parameter of two allows the incorrect diabetes code 02500110 as a candidate category; however, we avoid misclassifying "Hypertension" as "diabetes mellitus" because the MIN_EVENT_FREQ parameter set to 25 eliminates this code since the diagnostic entry frequency is five. Our experimental evidence suggests that such two-dimensional parameter tuning proves to be very effective, as discussed further.

Machine Learning Classification Component

If a given diagnostic statement is not successfully classified by example-based classification, the statement is submitted to the machine learning classification component. Classification with machine learning described in this article also relies on the data generated by the Medical Index staff over the past ten years, but does so in a different way from the example-based classification component. The machine learning component is trained on single words that comprise previously manually coded diagnostic statements. We use a sparse matrix implementation of the naïve Bayes algorithm, which happens to be a very robust and scalable approach for large amounts of textual data.

Naïve Bayes The Bayes decision rule chooses the class that maximizes its conditional probability given the context in which it occurs:

$$C' = \operatorname{argmax} P(C) \prod_{j=1}^{n} P(V_j|C) \quad (1)$$

Here, C' is the chosen category, C is the set of all categories, and V_j is the context. The naïve Bayes algorithm chooses the most likely category given a set of predictive features found in the context. The algorithm makes a simplifying assumption that the words in the context are independent of each other. In other words, the assumption is that if we see the word "heart" in some context then the word "edema" has as

much of a chance to be in the same context as the word "airplane." Clearly, this assumption is not true for human languages. Theoretically, such assumption makes naïve Bayes classifiers unappealing for text categorization problems, but in practice it turns out that the violation of the independence assumption has little effect on the accuracy of text categorization.[7, 22] It turns out that for binary classification problems, while the independence assumption is technically correct, a situation where two or more dependent features happen to predict different classes has a relatively low probability. A complete proof can be found in Domingos and Pazzani.[22] One of the major advantages of naïve Bayes as compared to other more sophisticated techniques is that it is robust, fast to train and does not require large amounts of computing resources.

"Bag-of-words" Data Representation In order to train the naïve Bayes classifier we represent each diagnostic statement as an unordered vector of features where the features are the words comprising the statement. This is known in machine learning literature as a "bag-of-words" technique. One of the challenging issues in processing clinical narratives as any other free text is the orthographic and lexical variability. The need for normalizing clinical text has been widely recognized and several approaches have been used to deal with issues such as stop word removal, word and sentence segmentation, spelling correction, stemming and abbreviation expansion.[23-25] None of these approaches are error-free. Due to stringent accuracy requirements on the data that will not be subsequently manually reviewed, we implemented a very conservative set of rules for text normalization. We exclude stop words such as "the," "a," "is," "was" (a total of 124 words). The input is also tokenized to treat certain multiword units as a single unit (e.g., words *grade, stage, level, phase, class, gravida, para, type, alpha, onset* followed by a Roman or an Arabic numeral) in addition to lowercasing all words that start with a capital letter followed by a lowercase letter. Words in all caps remain unchanged. We also remove some of the syntactic formatting such as the word "DIAGNOSIS:" in all caps at the start of the line.

Classifier Output and the Multiple Classification Problem- Once a naïve Bayes classifier is trained to associate words with categories, each new instance that is presented to the classifier needs to be translated into a "bag-of-words" feature vector in the same manner as was done with the training data. The classifier then computes a posterior probability for each category in the classification it was trained on. For example, if the classification has 20,000 categories used for training, then the classifier will compute a score for each category for the new input. The classifier then ranks scores in descending order so that the categories with the strongest association to the "bag-of-words" vector for a given diagnosis are found at the top of the list. This is not a problem if the data for each diagnostic statement have only one category; however, many of the diagnoses are assigned more than one HICDA code. This presents a considerable problem of determining how many of the top N categories produced by the classifier are the correct ones. This multiple category problem has been confronted in other domains such as biosurveillance from chief complaints[26] and automatic coding of responses to surveys,[27] as well as within the general framework of machine learning.[28]

We have explored two possible ways to address the multiple classification problem afforded by the SNoW implementation of the naïve Bayes classifier. The first is to train two classifiers: one to rank the active categories and the other to suggest how many of the top ranked categories to retain. The second is to represent the multiple categories assigned to a particular diagnostic statement as a single compound category. Both approaches have potential drawbacks. The former has two potential sources of inaccuracies instead of one, and the latter introduces a large number of new categories. We tested and evaluated both approaches and found the latter to be more beneficial on our specific task of classifying Mayo EMR diagnoses into HICDA despite its limited scalability problem. We have conducted a series of experiments with this method at the top level of the HICDA hierarchy, which has only 19 categories, and found that representing multiple classifier entries with composite categories is a promising approach.[21] We tested this approach on the leaf level of the hierarchy and report the results of the experiments further in this article.

Phrase Chunker Component The purpose of the phrase chunker is to split the incoming diagnostic statements into meaningful components and then attempt both example-based and machine learning classification on the individual constituents of the diagnostic statement. For example, "Myocardial infarction with subsequent emergency coronary artery bypass graft" would be chunked at the "with subsequent" divider. Just like in the overall architectural flow, the precedence is given to the codes identified with the example-based component. Thus, if a diagnostic entry consists of two codable elements such as "A and B" and the example-based component fails, then the phrase chunker will split this statement into constituents (A,B) and attempt the classification with the example-based component on each constituent individually and independently. So it is possible that A will be classified with the example-based component and B with the machine learning component and vice versa.

Results

The validity of both components of the Autocoder was evaluated using standard techniques of computing precision and recall. Precision and recall are measures widely used in the domains of machine learning and text categorization[29] and are defined in the "Evaluation Measures" subsection. The development of several reference standards is discussed in the "Reference Standard Development" section. Finally, the experiment results are reported in the "System Evaluation" section.

Evaluation Measures

We used the standard evaluation metrics of precision, recall and f-score. Precision is defined as the ratio of correctly assigned categories (true positives) to the total number of categories produced by the classifier (true positives and false positives).

$$Precision = \frac{tp}{tp + fp} \quad (2)$$

Recall is the ratio of correctly assigned categories (true positives) to the number of target categories in the test set (true positives and false negatives).

$$Recall = \frac{tp}{tp + fn} \quad (3)$$

F-score represents the harmonic mean of precision and recall according to the formula in (4):

$$F - score = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} \quad (4)$$

where P is the precision, R is the recall and α is a weight that is used to favor either precision or recall. In our computations α was set to 0.5 indicating equal weight given to precision and recall.

Two sets of precision/recall results are reported: micro-averaged and macro-averaged as described in Manning and Shutze.[29] The micro-averaging method represents the results where true positives, false positives and false negatives are added up across all test instances first and then these counts are used to compute the statistics. The macro-averaging method computes precision/recall for each test instance first, and then averages these statistics over all instances in the reference standard. These two methods yield different results when the instances have more than one correct category and when categories are represented by unequal numbers of instances. The micro-averaging method favors large categories with many instances, while the macro-averaging method shows how the classifier performs across all categories.[29]

Training and Testing Data

Three reference standards were developed to evaluate the Autocoder. Given the architecture of the Autocoder and the data flow logic built into it, each diagnostic statement that enters the system can fall into three broad categories. We will refer to them as A, B and C. The first type (A) consists of statements that have passed the example-based component's filter controlled by the MIN_EVENT_FREQ parameter set at 25. We can classify these data solely with the example-based component and with high confidence—the categories assigned to this type will not be subsequently reviewed.

The second type of data (B) is made up of diagnostic statements that have been found in the Medical Index database of previously coded examples, but whose diagnosis-gender-code event frequency is lower than the value of the MIN_EVENT_FREQ parameter set at 25. We are less confident in classifying a case like this and therefore submit this case for manual review.

The third type (C) consists of diagnostic statements of which we do not have any prior record. These types of data need to be classified with the machine learning component. The codes assigned to these diagnostic statements are of low confidence and can only be used as suggestions for subsequent manual review.

All available data samples collected between 1994 and 2004 are split into training and testing according to the flow diagram in Figure 3. The training data consisted of over 22 million non-unique examples entered into the database between 1994 and June 1, 2003. The testing data consisted of 898,584 examples collected between June 1, 2003 and January 1, 2004. In order to determine the distribution of the three types of data we looked up each testing data sample in the database created from the training data samples and determined if it belonged to one of the following three types: A, B, or C. There were 527,673 samples (58.7%) of type A data, 213,440 samples (23.7%) of type B data, and 157,471 samples (17.5%) of type C data. We drew several random samples from each of the three datasets to create reference standards as discussed in the following subsections.

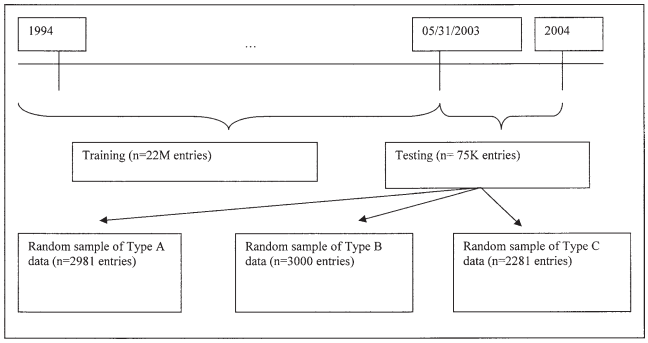

Figure 3. Training and testing data collection schedule.

Pilot Studies

In order to create a reference standard with acceptable statistical power we conducted a pilot study to determine the expected level of precision and recall and to optimize the parameters of the example-based component. We created a random sample of 75,000 entries from the type A data set. A regression test for precision/recall of the example-based component was performed by varying two parameters: MIN_EVENT_FREQ and MAX_NUM_CAT. MIN_EVENT_FREQ was varied in the range between 1 and 200 (Figure 4) and the MAX_NUM_CAT parameter between 1 and 10 (Figure 5).

Initially, we varied the MAX_NUM_CAT parameter while holding the MIN_EVENT_FREQ parameter steady at its lowest value of 1. This was the logical starting point as we knew that the majority of diagnostic statements are assigned either one, two, or three codes. Figure 6 shows the actual frequency distribution of diagnoses with various code assignments. The distribution drops off sharply after three categories per diagnostic statement; however, we did extend the variation of the parameter to 10. The results in Chart 2 show that there actually is a point at which the precision and recall curves cross. When the parameter is changed from 1 to 2, the recall goes up from 96.1% to 97.6% while the precision drops from 97.4% to 96.2%. When the MAX_NUM_CAT parameter is set to 3 or higher, the recall stays about the same; however, the precision drops dramatically, as is expected. This result allows us to optimally set MAX_NUM-_CAT parameter to 2.

Once the optimal value for MAX_NUM_CAT parameter was determined, we optimized the MIN_EVENT_FREQ parameter by holding MAX_NUM_CAT steady at its lowest value. The results in Chart 1 show that the most optimal MIN_EVENT_FREQ value is 25. Despite the fact that the precision and recall curves do not cross, it is clear that the growth in precision asymptotes at MIN_EVENT_FREQ set to 25. Not much is gained in recall in going lower than 25, but there is a substantial drop in precision; therefore 25 is set as the most optimal value for MAX_EVENT_FREQ.

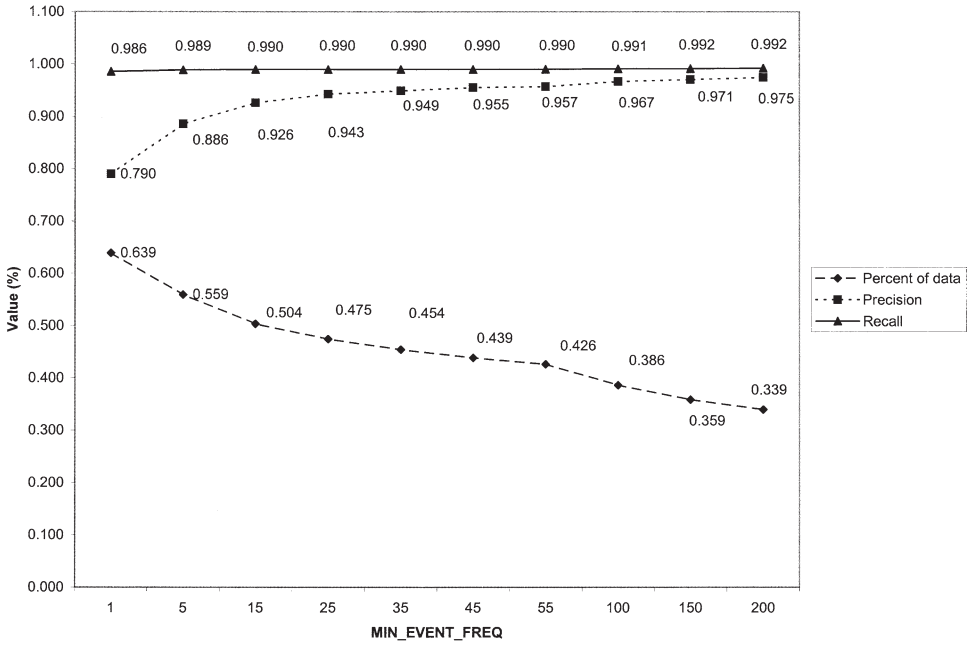

Figure 4. Precision/Recall results where MIN_EVENT_FREQ parameter is varied between 1 and 200 and MAX_CAT_NUM is held at 1.

Using both MAX_NUM_CAT set to 2 and MAX_EVENT_FREQ set to 25, we arrive at 97% precision and 94% recall on the test set of 75,000 instances. According to our statistical power calculations, at this level of precision and recall we would need to examine over 2,600 random samples manually in order to estimate the results within a 1% margin of error using a 95% confidence interval.

Type A Reference Standard

We compiled a set of 3,000 entries from a sample that had been coded manually by the Medical Index staff as shown in Figure 3. These entries were manually re-verified for accuracy and completeness by two senior Medical Index staff with more than ten years of medical classification experience. Nineteen instances were excluded due to technical problems such as missing text of the entry or patient gender information. The resulting set of 2,981 instances was used as the reference standard for further evaluations of the example-based component.

Type B Reference Standard

A random sample of 3,000 entries with frequency below 25 was extracted from the same test set of 75,000 instances used to

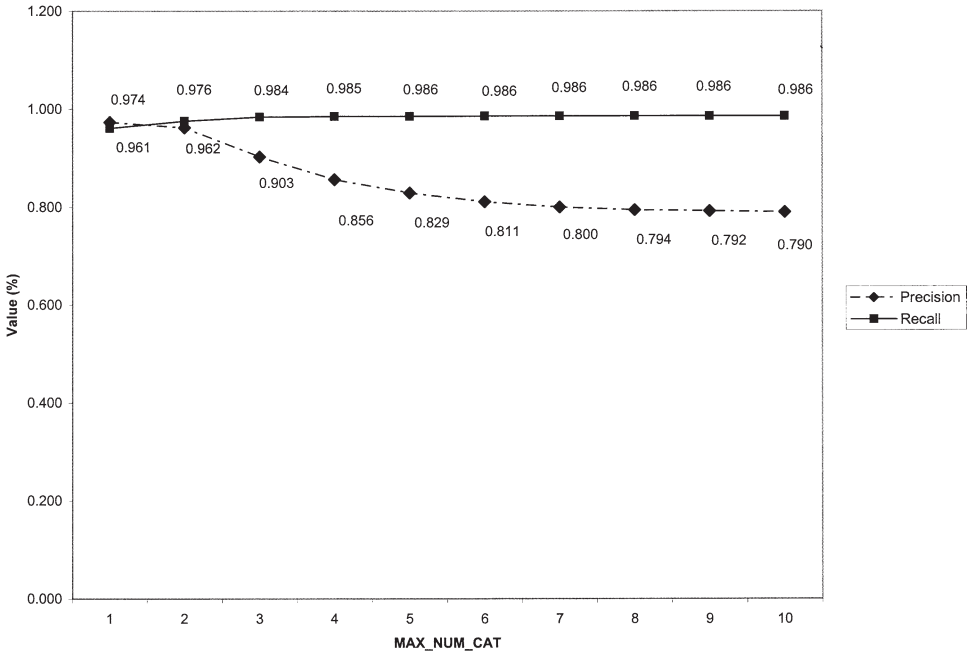

Figure 5. Precision/Recall results where MAX_CAT_NUM parameter is varied between 1 and 10 and MIN_EVENT_FREQ is fixed at 1.

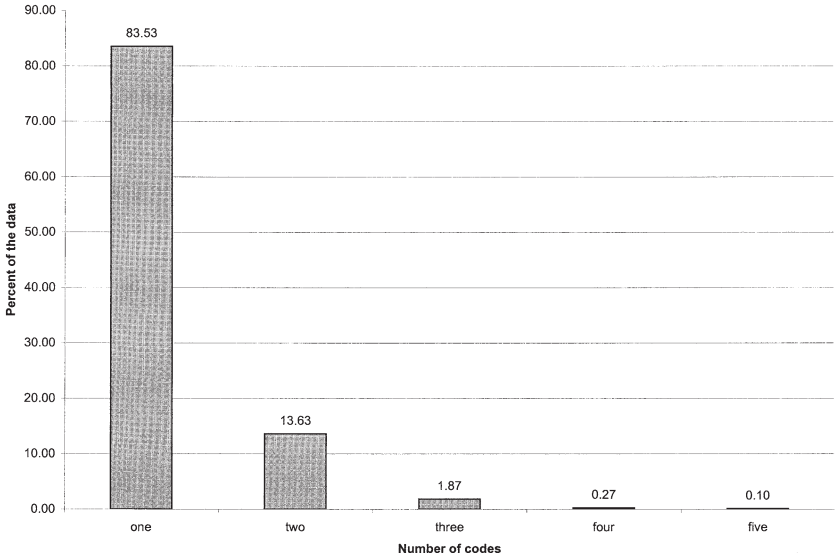

Figure 6. Distribution of the number of codes assigned to diagnoses in the test data.

develop the type A reference standard. This population is complementary to the population used to sample type A reference instances due to the single frequency threshold.

Type C Reference Standard

We compiled a random sample of 3,000 entries that were not found via lexical string match in the database used to train the example-based and the machine learning components. We wanted to make sure that the entries in this set were truly never seen before and used a more aggressive normalization as well as random manual checking. This resulted in a set of 2,281 entries in the final reference standard.

Neither type B nor type C entries were manually re-verified by classification experts; however, both of these types of entries had been manually classified before and thus can serve as a reference standard. We are confident that these data are of high quality because the standard manual coding process involves initial coding with subsequent verification by a more experienced coder. Thus we can think of type A data as being doubly verified. The agreement between the first and second verification on type A data set is 94%. This provides an empirical foundation for our confidence in the quality of coding on type B and type C data and obviates the need for additional verification.

System Evaluation

The Autocoder was evaluated on the three reference standards, with each component of the Autocoder having been evaluated on the appropriate standard. The evaluation results for all three types are presented in Table 1 and Table 2.

Evaluation on Type A Data

Type A data consist of diagnostic entries found in the database of previously coded entries with frequency greater than or equal to an empirically established threshold of 25. Since the example-based classifier component is intended to operate without subsequent review, it was necessary to optimize the parameters to maximize precision and recall as well as its capture rate (the number of entries processed). Since MAX_NUM_CAT parameter does not affect the capture rate, we plotted the capture rate with respect to the variation in the MIN_EVENT_FREQ only in Chart 1. With MIN_EVENT_FREQ set at 25, we are able to capture 47.5% of the unique test entries.

With MAX_NUM_CAT parameter set to 2 and MIN_EVENT_FREQ set to 25, the Autocoder achieved a precision of 96.7% and recall of 96.8% resulting in an f-score of 96.7 using the micro-averaging method and a precision of 98.0% and recall of 98.3% (with an f-score 98.2) using the macro-averaging method.

Evaluation on Type B Data

Type B data consist of diagnostic entries found in the database of previously coded entries with frequency lower than an empirically established threshold of 25. Diagnostic statements classified as type B were categorized using the example-based component and were sent for subsequent manual review. The micro-averaging method yielded a precision of 86.6%, recall of 93.7%, and an f-score of 90.4, while the macro-averaging method yielded a precision of 90.1%, recall of 95.6% recall, and an f-score of 93.1.

Table 1 ■ Micro-average Precision/Recall Results for Type A, B and C Data. The Cells for Precision Recall Contain the Number of True Positives (tp) Followed by the Sum of True Positives (tp) and False Positives (fp) or False Negatives (fn), Followed by the Percentage Value, Followed by the Width of a 95% Confidence Interval

	Precision tp/tp + fp (%)	95% CI	Recall tp/tp + fn (%)	95% CI	F-score
Type A data	3,514/3,630 (96.7)	±0.5	3,514/3,628 (96.8)	±0.5	96.7
Type B data	3,777/4,361 (86.6)	±1.0	3,777/4,028 (93.7)	±0.7	90.4
Type C data	1,663/2,834 (58.6)	±1.7	1,663/3,733 (44.5)	±1.6	50.7

Table 2 ■ Macro-average Precision/Recall Results for Type A, B and C Data. The Cells for Precision Recall Contain the Macro-averaged Value for Precision/Recall, Followed by the Total Number of Test Instances, Followed by the Width of a 95% Confidence Interval

	Precision avg. precision (N samples)	95% CI	Recall avg. precision (N samples)	95% CI	F-score
Type A data	98.0 (2,981)	±0.5	98.3 (2,981)	±0.4	98.2
Type B data	90.1 (3,000)	±0.7	95.6 (3,000)	±0.6	93.1
Type C data	58.5 (2,218)	±1.9	50.7 (2,218)	±1.8	54.4

Evaluation on Type C Data

Type C data consist of diagnostic entries not found in the database of previously coded entries. The best results for this data type are displayed in Table 1 and 2. The micro-averaged technique yielded a precision of 58.6%, recall of 44.5%, and an f-score of 50.7. The macro-averaged technique yielded a precision of 58.5%, recall of 50.7%, and an f-score of 54.4.

Discussion

The authors have shown that the example-based component achieved precision/recall results that exceeded our objectives and were deemed appropriate to be left unsupervised by manual verification. With the current parameter settings, this component is able to process 48% of all unique physician generated diagnostic statements at the Mayo Clinic without a need for subsequent review. In practical terms, this capture rate of 48%, computed for unique diagnostic strings, is likely to be an underestimate as some of the "easier" to code diagnoses such as "hypertension" also happen to be highly recurrent. The type A test set of 3,000 diagnoses does not reflect the individual frequencies of the diagnoses and thus produces a lower bound estimate of the capture rate. The rate on non-unique entries of type A is 59%.

Furthermore, the performance on type B data, which comprised an additional 24% of the non-unique entries, could be classified with only a slightly lower recall and precision than the performance on type A data. While the accuracy is not high enough to justify eliminating subsequent manual review, it will aid the coding process. Although the performance on type C data is much lower than type A or type B, this type of data comprised only 18% of the non-unique diagnoses entered into the system. This is consistent with Gundersen et al.[17] where they found that their system could not produce encodings on 15% of the diagnoses. It is unclear at this point whether providing codes with the naïve Bayes classifier at 60% precision and 50% recall is at all beneficial in practical terms to expedite the manual review. Further usability studies are necessary to determine this. So far, our validation study shows that we can reliably achieve our design objective, which is to increase the throughput of the Medical Index staff without any significant loss of coding accuracy at least on 82% of the incoming diagnoses.

Several areas for future improvement were identified. One such area is data representation for the prediction of the number of codes. The "bag-of-words" approach we used to represent data for the classifier that predicts the number of codes is probably suboptimal. We believe that in order to improve on this classification task, we need to take into account such features as the number of clauses, phrases, and individual words in each sample as well as presence or absence of some of the clear orthographic clues such as commas, periods, and semicolons along with the lexical content of the samples. The latter should be helpful in predicting the correct number of codes for neoplasms, which are almost always assigned to at least two categories (malignancy and location), while the former should help with the samples that contain multiple coded entities.

From the standpoint of wide applicability of this research, one has to address the issue of HICDA representing an outdated version of ICD. Our methodology and software can be extended to work with other categorization schemes, provided that these schemes have been used in medical coding practices and collected substantial amounts of manually coded textual data. Of course, the amount of effort required to extend this methodology will depend on the complexity and the specifics of the classification to which it is extended. Currently, a mapping table exists that can be used to convert HICDA codes into ICD-9 codes and subsequently into SNOMED-CT codes, albeit with a large loss in granularity.

Limitations

There are several known limitations in the design of the Autocoder. The first limitation is the fact that diagnostic statements represent only a part of a patient's electronic medical record, which introduces a limitation on the accuracy of any classifier trained solely on the information present in the diagnostic string. For example, a diagnostic statement of "dementia" taken in isolation from the rest of the note is ambiguous. It can be coded as "Alzheimer's disease" or "Dementia, NOS" depending on the age of the patient as well as other factors not reflected in the diagnostic statement itself. To overcome this limitation, a more complete approach to clinical note classification would have to involve representations of other segments of the record.

Another limitation is the reference standards. In the process of creating and re-verifying the reference standard of type A, we were able to identify a number of inherently ambiguous categories whose correct choice depends on the age of the patient. Since age is a continuous value, we would need a systematic way of assigning a discrete value to this variable in order to be able to use it as a feature in classification. So far, we were unable to determine a systematic and reliable way of assigning patient's age to a discrete value; therefore, we set the age dependent categories aside into a special table so that if the example-based component produces a code that happens to be in that table, we would mark that entry as requiring a subsequent review. This limitation may affect the number of diagnostic entries that can be autocoded without subsequent review; however, we do not believe this

effect would be significant. We found only 43 codes out of nearly 30,000 that were age-dependent. Only 38 entries out of the 3,000 in type A data set contained one or more of these codes.

Finally, it is important to note that due to resource limitations the sample size of the reference standard is fairly small compared to the universe of all diagnostic statements. The results obtained with this reference standard should be interpreted with caution—they are generalizable to the more frequent diagnoses and diagnostic categories but probably not the ones that are relatively rare.

Conclusion

We have presented a system for automatic classification of clinical diagnoses that appear as part of clinical notes at the Mayo Clinic. Our system has the advantage of relying on the knowledge base obtained by having over ten years of manual coding experience. The system is designed to make manual classification of clinical diagnostic entries more efficient in terms of both throughput and accuracy by using previously manually coded examples to train the various classification components of the system. Over two-thirds of all diagnoses are coded automatically with high accuracy. Of these, approximately half of the diagnoses are automatically coded with precision and recall over 95% and will not be reviewed manually. The Autocoder has been successfully implemented, which resulted in a reduction of staff engaged in manual coding from thirty-four coders to seven verifiers. Further development and validation of this technology will be necessary in order to maximize its effectiveness.

References ■

1. Kurland LT, Molgaard C. The patient record in Epidemiology. Scientific American 1981;245(4):63–65.
2. Melton L. History of the Rochester Epidemiology Project. Mayo Clinic Proc 1996;7(3):266–74.
3. H-ICDA Hospital Adaptation of ICD-A-HICDA (second edition). 2nd ed. Ann Arbor, MI: CPHA (Commission on Professional and Hospital Activities), 1973.
4. Yang Y, Chute CG. A Linear Least Squares Fit mapping method for information retrieval from natural language texts. In: 14th Int Conf Comput Ling (COLING 92), 1992; Tokyo, Japan. 1992:447–53.
5. Joachims T. Text categorization with support vector machines: Learning with many relevant features. In: Proc 10th Eur Conf Mach Learn: Springer-Verlag. 1998:137–42.
6. Wilcox A, Hripcsak G. The role of domain knowledge in automating medical text report classification. J Am Med Inform Assoc 2003;10:330–8.
7. Lewis D. Naive (Bayes) at forty: The independence assumption in information retrieval. In: 10th Eur Conf Mach Learn (ECML 98); 1998; Chemnitz, Germany. 1998:4–15.
8. Johnson D, Oles F, Zhang T, Goetz T. A decision-tree-based symbolic rule induction system for text categorization. IBM Syst J 2002;41(3):428–37.
9. Nigam K, Lafferty J, McCullum A. Using Maximum Entropy for text classification. In: Workshop on Machine Learning for Information Filtering (IJCAI 99); 1999; Stockholm, Sweden. 1999: 61–7.
10. Aronow D, Soderland S, Ponte J, Feng F, Croft W, Lehnert W. Automated classification of encounter notes in a computer based medical record. In: Medinfo; 1995; Vancouver, Canada. 1995:1–8.
11. Aronow D, Fangfang F, Croft B. Ad hoc classification of radiology reports. J Med Inform Assoc 1999;6(5):393–411.
12. Aronsky D, Haug P. Automatic identification of patients eligible for a pneumonia guideline. In: American Medical Informatics Association Symposium (AMIA); 2000; Los Angeles, CA. 2000: 12–6.
13. Fiszman M, Chapman WW, Aronsky D, Evans S, Haug P. Automatic detection of acute bacterial pneumonia from chest X-ray reports. J Am Med Inform Assoc 2000;7(6):593–604.
14. Friedman C, Alderson PO, Austin JH, Cimino JJ, Johnson SB. A general natural-language text processor for clinical radiology. J Am Med Inform Assoc 1994;1(2):161–74.
15. Wilcox A. Automated Classification of Text Reports [Ph.D. Thesis]: Columbia University, NY; 2000.
16. Payne TH, Gaster B, Mineer D, et al. Creating a note classification scheme for a multi-institutional electronic medical record. In: American Medical Informatics Association Fall Symposium; 2003 November 10; Washington, DC; 2003.
17. Gundersen ML, Haug PJ, Pryor TA, et al. Development and evaluation of a computerized admission diagnoses encoding system. Comp Biomed Res 1996;29(5):351–72.
18. Yang Y. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: 17th Ann Int ACM SI-GIR Conference on Research and Development in Information Retrieval (SIGIR'94); 1994; Dublin, Ireland. 1994: 13–22.
19. Chute CG, Yang Y. An evaluation of computer-assisted clinical classification algorithms. J Am Med Inform Assoc 1994; 18(Symp. Suppl.):162–6.
20. Yang Y, Chute CG. An application of Expert Network to clinical classification and MEDLINE indexing. J Am Med Inform Assoc 1994;18(Symp. Suppl.):157–61.
21. Pakhomov S, Buntrock J, Chute CG. Using compound codes for automatic classification of clinical diagnoses. In: MedINFO; 2004; San Francisco, CA. 2004:411–5.
22. Domingos P, Pazzani M. On the optimality of the simple Baysean classifier under zero-one loss. Mach Learn 1997;29(2-3):103–30.
23. Travers DA, Haas SW. Using nurses' natural language entries to build a concept-oriented terminology for patients' chief complaints in the emergency department. J Biomed Inform 2003;36: 260–70.
24. Friedman C. A broad-coverage natural language processing system. In: American Medical Informatics Association Symposium; 2000 November 4-8; Los Angeles, CA. 2000:270–4.
25. Sager N, Lyman MS, Bucknall C, Nhan NT, Tick LJ. Natural Language Processing and the Representation of Clinical Data. J Am Med Inform Soc 1994;1(2):142–60.
26. Chapman WW, Christensen LM, Wagner MM, et al. Classifying free-text triage chief complaints into syndromic categories with natural language processing. Art Intell Med 2005;33(1):31–40.
27. Giorgetti D, Sebastiani F. Multiclass text categorization for automated survey coding. In: 18th ACM Symposium on Applied Computing; 2003; Melbourne, US: ACM Press, New York, US. 2003:798–802.
28. Har-Peled S, Roth D, Zimak D. Constraint classification for multiclass classification and ranking. In: The Conference on Advances in Neural Information Processing Systems (NIPS); 2003 Dec. 9-10; Vancouver, Canada: MIT Press. 2003:785–92.
29. Manning C, Shutze H. Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press; 1999.