Combining Bayesian Text Classification and Shrinkage to Automate Healthcare Coding: A Data Quality Analysis

EITEL J. M. LAURÍA, Marist College, Poughkeepsie, NY
ALAN D. MARCH, Hospital Universitario Austral, Buenos Aires, Argentina

This article analyzes the data quality issues that emerge when training a shrinkage-based classifier with noisy data. A statistical text analysis technique based on a shrinkage-based variation of multinomial naive Bayes is applied to a set of free-text discharge diagnoses occurring in a number of hospitalizations. All of these diagnoses were previously coded according to the Spanish Edition of ICD9-CM. We deal with the issue of analyzing the predictive power and robustness of the statistical machine learning algorithm proposed for ICD-9-CM classification. We explore the effect of training the models using both clean and noisy data. In particular our work investigates the extent to which errors in free-text diagnoses propagate to the classification model. A measure of predictive accuracy is calculated for the text classification algorithm under analysis. Subsequently, the quality of the sample data is incrementally deteriorated by simulating errors in the text and/or codes. The predictive accuracy is recomputed for each of the noisy samples for comparison purposes. Our research shows that the shrinkage-based classifier is a valid alternative to automate ICD9-CM coding even under circumstances in which the quality of the training data is in question.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; I.5.4 [**Pattern Recognition**]: Applications—*Text processing*; I.5.1 [**Pattern Recognition**]: Models—*Statistical*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classified design and evaluation*

General Terms: Algorithms

Additional Key Words and Phrases: ICD9-CM, automated coding, text classification, shrinkage, data quality

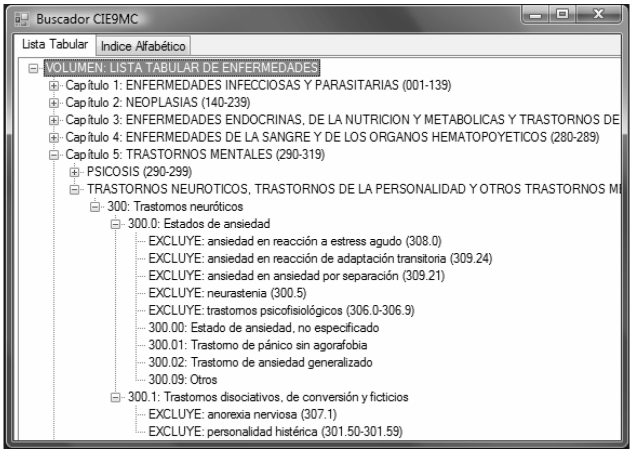
ACM Reference Format:
Lauría, E. J. M. and March, A. D. 2011. Combining Bayesian text classification and shrinkage to automate healthcare coding: A data quality analysis. ACM J. Data Inform. Quality 2, 3, Article 13 (December 2011), 22 pages.
DOI = 10.1145/2063504.2063506 http://doi.acm.org/10.1145/2063504.2063506

1. INTRODUCTION

In order to benefit the most from Information Technology (IT), medical production environments require that vast amounts of information produced during physician-patient encounters, diagnostic testing, and therapeutic procedures be made readily available to computer systems. Most of the information in clinical scenarios is recorded as free text in narrative form, which is prone to typographical errors and misinterpretations of ambiguous terms and phrases. Consequently, improving (or indeed rendering) machine readability of available free-text information remains the centerpiece of the problem. For solving this issue, researchers have resorted to the manual coding of information contained in medical documents, using a wide variety of coding schemes.

Author's address: E. J. M. Lauría, Marist College, Poughkeepsie, NY; email: eitel.lauria@marist.edu.

DOI 10.1145/2063504.2063506 http://doi.acm.org/10.1145/2063504.2063506



- ⊟ VOLUMEN: LISTA TABULAR DE ENFERMEDADES
  - ⊞ Capítulo 1: ENFERMEDADES INFECCIOSAS Y PARASITARIAS (001-139)
  - ⊞ Capítulo 2: NEOPLASIAS (140-239)
  - ⊞ Capítulo 3: ENFERMEDADES ENDOCRINAS, DE LA NUTRICION Y METABOLICAS Y TRASTORNOS DE
  - ⊞ Capítulo 4: ENFERMEDADES DE LA SANGRE Y DE LOS ORGANOS HEMATOPOYETICOS (280-289)
  - ⊟ Capítulo 5: TRASTORNOS MENTALES (290-319)
    - ⊞ PSICOSIS (290-299)
    - ⊟ TRASTORNOS NEUROTICOS, TRASTORNOS DE LA PERSONALIDAD Y OTROS TRASTORNOS M
      - ⊟ 300: Trastornos neuróticos
        - ⊟ 300.0: Estados de ansiedad
          - EXCLUYE: ansiedad en reacción a estress agudo (308.0)
          - EXCLUYE: ansiedad en reacción de adaptación transitoria (309.24)
          - EXCLUYE: ansiedad en ansiedad por separación (309.21)
          - EXCLUYE: neurastenia (300.5)
          - EXCLUYE: trastornos psicofisiológicos (306.0-306.9)
          - 300.00: Estado de ansiedad, no especificado
          - 300.01: Trastorno de pánico sin agorafobia
          - 300.02: Trastorno de ansiedad generalizado
          - 300.09: Otros
        - ⊟ 300.1: Trastornos disociativos, de conversión y ficticios
          - EXCLUYE: anorexia nerviosa (307.1)
          - EXCLUYE: personalidad histérica (301.50-301.59)

Fig. 1. Spanish ICD9-CM navigator. The successive levels of the hierarchy are easily identified as Volume, Chapter, Section, and 3-, 4- and 5-digit level codes.

The problem arises from the fact that classification systems such as ICD-9-CM[1] appear to be simple code lists but are really complex rule-based coding systems [Rector 1999].

ICD-9-CM is a refinement of the World Health Organization's International Classification of Diseases and Related Health Problems (commonly known as ICD in healthcare environments). ICD provides a set of codes which allows for indexing diseases as well as a signs, symptoms, general health complaints, motives for seeking medical care, and external causes of injury or disease. ICD is used worldwide for assembling morbidity and mortality statistics for epidemiological purposes. ICD-9-CM represents a US-developed refinement of WHO's ICD, which provides greater clinical detail by adding an additional level to the leaf (bottom level) categories of the latter. This additional level is commonly known as the fifth digit, as the digits in the code reflect the position in the hierarchy, and ICD-9-CM provides an extra digit (an additional level of detail). The codes that are effectively assigned to-diagnosis may be aggregated into blocks of decreasing levels of granularity, thus forming a tree-like structure similar to the Internet newsgroup hierarchies. The successive levels of the hierarchy are easily identified as Volume, Chapter, Section, and 3-, 4- and 5-digit level codes.

One particular aspect of ICD-9-CM is the "ragged" nature of its tree structure, meaning that the classification is not systematic regarding the number of nested categories which may be found underlying each root class. Thus, in some cases leaf nodes appear at a 4th level, whereas in other cases they appear at a 6th level.

ICD-9-CM is extensively used throughout the US both for statistical and reimbursement purposes and is compulsory for healthcare providers working for agencies such as Medicare. It has also been adopted (sometimes through local adaptations) by other countries (e.g., Spain, Australia). Figure 1 shows a screenshot of the computerized ICD-Navigator used by human coders, which illustrates the hierarchical organization of ICD-9-CM.

Manual ICD-9-CM coding is a laborious and cost-consuming process that requires specially trained human resources [March et al. 2004]. The huge volume of information generated by healthcare production environments imposes a restriction on the

[1] International Classification of Diseases, Ninth Revision, Clinical Modification (http://www.cdc.gov/nchs/icd9.htm).

feasibility of coding all the information in a cost-efficient and timely manner. For these reasons several authors have explored the possibility of automating the coding process, framing the problem as a text classification activity [Chapman et al. 2005; Chute et al. 1990; Cooper and Miller 1998; Evans et al. 1996]. Different techniques [Friedman et al. 2004; Goldstein et al. 2007; Lussier et al. 1999; Mykowiecka et al. 2009] have been considered to fulfill this task. Although these techniques have met with a certain level of success, the fact that most of them rely on grammar-based rules has precluded their application in circumstances that go beyond the original settings where these techniques were originally used. In this sense, language barriers pose the greatest obstacles to reproducibility, as they often require extensive rewriting of algorithms [March et al. 2004]. Supporters of machine learning strategies have pointed out deficiencies in the formal grammars approach, such as the expensive and time-consuming character of assembling grammars and dictionaries for different domains or their failure to deal with the ambiguities of language and ungrammaticality of everyday speech.

The automated classification of free-text documents is a classic statistical machine learning problem: A learning model is created using an algorithm and a training set of free-text samples, each of them labeled with a given document class value. The trained model is then tested using a collection of labeled samples to verify the accuracy of the classification method. Nevertheless, in the area of medical informatics, statistical machine learning applications of text classification are still scarce and the authors' preliminary reports are among the few published [Goldstein et al. 2007; Lauría and March 2006, 2007; Lauría and Tayi 2003; March et al. 2004; Suominen et al. 2008; Tremblay et al. 2009].

Medical coding (e.g., ICD coding) usually implies assigning medical diagnosis to specific class values (medical codes) extracted from a very large pool of categories. It follows that an automated statistical classifier will typically require a very large training dataset to render accurate results, and such a large volume of training data may not be readily available.

Also, when dealing with statistical machine learning approaches for automated coding, the quality of the input data used for training purposes becomes an item of concern. Statistical machine learning algorithms rely on training datasets to develop statistical models based on which predictions and inferences are made. Due to this fact, the quality of the input data has a direct impact on the predictive accuracy of the algorithms, and the effective use of these techniques requires that the input data attains a certain degree of quality. There is a trade-off between the cost of guaranteeing input data quality and the cost of misclassification given by the inadequate predictive accuracy of the models developed with the input data at hand.

Based on these observations, the experimental focus of this article is on the analysis of a text classification algorithm for automated ICD9-CM coding using training data of moderate size and varying quality. The specific aim of this work is to analyze the robustness of the shrinkage-based classifier used for automatic ICD9-CM coding, under circumstances in which data quality is questionable. Real-world applications such as healthcare coding necessarily involve use of data whose quality varies and can therefore be adequate on some dimensions and poor on others. Thus the efficacy and usability of a statistical machine learning technique could be strongly affected by the quality of data available to a healthcare organization. The persistent quality issues of data originated and consumed in organizational settings are well documented [Tayi and Ballou 1998]. The specialized literature categorizes data quality across a number of dimensions including accuracy, completeness, interpretability, consistency, and timeliness [Strong et al. 1997]. So a timely and complete dataset may still be considered of poor quality if it contains inaccurate data. In cases such as machine learning-based text classification, the quality of the training data is intrinsic to the data itself [Fisher et al. 2006], meaning that a data quality enhancement process is a prerequisite to any successful statistical machine learning application [Lauría and Tayi 2003]. However, determining the scope of such data quality enhancement activity is not straightforward, as it requires an understanding of the interplay of several concurrent factors: the nature of data errors, the intrinsic characteristics of the problem domain, and the proposed machine learning technique [Lauría and Tayi 2003]. In this work we carry out a study of the performance of the shrinkage-based classifier.

We have done preliminary research applying Bayesian classifiers to the task of automated ICD9-CM coding using a computerized version of the 1999 Spanish Edition of the named classification scheme, thus producing a set of codes univocally related to one or more of the section codes occurring for each hospitalization [Lauría and March 2006; March et al. 2004]. Our work extends this preliminary exploration by focusing on a shrinkage-based approach and considering a much larger training dataset to develop the statistical models and rigorously test its predictive accuracy in comparison with other state-of-the-art classifiers using training data with varying degrees of quality.

The following section provides a brief literature review of automated medical coding. Then the article proceeds with a short primer on the shrinkage-based variation of multinomial naive Bayes, the text classification algorithm used throughout the experimental setup of the article. The next section describes the experiments and reports the results. The article ends with a discussion and our conclusions, including future research pointers.

2. AUTOMATED MEDICAL CODING

One of the main problems with manual medical coding is that it is a time-consuming and expensive process requiring specially trained human resources. A leading healthcare informatics industry journal has qualified the current (manual) coding workflow as “expensive and inefficient” [AHIMA 2004]. Its bibliography is ripe with a variety of reasons determining the low precision of manual coding [Stausberg et al. 2008]. The problem is further confounded by the fact that many different medical coding systems are available [Strang et al. 2002], each seeking to satisfy different classification needs and requiring special training or software for adequate deployment (e.g., DRGs)[2]. In many cases, more than one coding system must be applied to a given dataset in order to comply with national (DRGs and/or ICD-9 in the US) or international regulations (ICD-10 in Europe and South America). Due to the lack of interoperability, in many cases cross-coding (transformation of one given set of codes into another set belonging to a different coding system) is only partially possible. Briefly stated, coding information into any given coding system is expensive in itself, and costs usually rise due to the fact that more than one coding system might be required for certain environments.

A number of researchers have suggested computer-based automation of medical coding (also known as autocoding or computer-assisted coding) using natural language processing (NLP) techniques [Chapman et al. 2005; Friedman et al. 2004; Rassinoux et al. 1994]. Generally speaking, and described in greater detail in the following section, NLP Techniques may be classified as pertaining to one of two main approaches: grammar (or rule)-based and machine learning-based. Most research in the area of autocoding and medical NLP has employed the former, either alone or on occasions combined with some form of machine learning-based NLP. In former publications, the authors of this proposal have successfully employed pure statistical approaches and intend to further pursue this line of research in the future.

[2] Diagnosis-Related Groups (See R. B. Fetter, D. A. Brand, and D. Gamache Eds., *DRGs, Their Design and Development*, Health Administration Press, Ann Arbor [1991]).

Our research supports the notion that free text may be rendered machine understandable by applying NLP techniques. *Machine readability*, in the context of this work, should be understood as the capacity to automatically translate "chunks" of free texts into their conceptual representation as a given code in a given coding system. Following our proposed technique, a learning algorithm is exposed to a training set of hand-coded free-text discharge diagnoses, based on which it learns to classify new sets of texts (test sets) to which it will be duly exposed. Our intent is to measure the reliability of the proposed algorithm and its robustness to deal with training data of poor quality.

Our work uses Spanish language discharge diagnoses in order to render their corresponding codes, which are available both in Spanish and in English versions. Given that the original diagnoses are in Spanish, and that resulting codes have English language descriptors, the original Spanish texts could be made available to non-Spanish speakers. In the context of our work, we measure the performance of text classification algorithms for which little or no experience is available for non-English languages. Areas of application include bulk processing of medical information produced in non-English speaking countries but that may be of interest in their English counterparts. This application could prove useful for processing medical information for issues relating to immigration, employment, and homeland security.

2.1 Formal Grammar and Machine Learning Approaches to NLP

The fundamental enabling technology for automatic coding of medical text is natural language processing (NLP). Techniques for NLP may be classified into two major groups: formal grammar-based and machine learning-based. Briefly stated, formal grammar approaches are generally based on the discovery and utilization of rules that apply to well-constructed discourse. Some of its methods are procedural and based on state machines where a typical model is comprised of states, and transitions among states, with a text for an input and an array of marked-up words or parts of speech as an output. Other formal approaches take advantage of declarative computer languages such as Prolog or Lisp to reproduce the formal rules known to apply to well-constructed discourse [Héja et al. 2007]. Both procedural and declarative approaches seek to build and apply formal models such as regular grammars, context-free grammars, and feature-augmented grammars. Formal grammar approaches draw much of their methods from classical linguistics and make a strong use of techniques based on the morphological, syntactic, semantic, and pragmatic aspects of natural language [Chierchia and McConnell-Ginet 1990; Chomsky 1965; Cruse 1986; Moreno 1998; Mykowiecka et al. 2009]. Generally speaking, formal approaches are mainly qualitative.

Machine learning-based NLP adopts a different approach by initially ignoring the fact that people construct text in accordance with structural rules. One of the main approaches of machine learning techniques is probabilistic and may be basically summed up in the notion that words do not occur randomly in a given text but instead are the subject of a probability function [Joachims 1997; Mccallum et al. 1998; Eyheramendy et al. 2003]. In the context of automated coding, and from a statistical point-of-view, an extremely basic example of this approach would be the fact that the free-text expression "Type II diabetes with incipient retinopathy" is associated with ICD9-CM code 250.00 with a probability of .93.

An impressive amount of research in the area of the application of NLP to medical informatics has utilized formal grammar approaches. In the case of automated mapping of medical texts to codes, researchers have resorted to different standardized coding systems such as UMLS [Chute et al. 1990; Evans et al. 1996; Friedman et al. 2004; Hersh et al. 2001; Lowe et al. 1999; Mendonca et al. 2005; Nadkarni et al. 2001;

Zou et al. 2003], MESH [Cooper and Miller 1998; Elkin et al. 1988; Moore et al. 1987], ICD-9-CM [Lussier et al. 1999], SNOMED [Lussier et al. 2001] and others [Friedman et al. 2004], as well as ad hoc terminologies [Hahn et al. 2002; Hripcsak et al. 1995, 2002; Lin et al. 1991; Rassinoux et al. 1994; Zelingher et al. 1995]. Compared to the formal approach, machine learning-based NLP applied to medical free text has seen much less development. Some authors have extended formal grammar-based models with probability functions [Chapman et al. 2005; Christensen et al. 2002; Cooper and Miller 1998]. "Pure" machine learning methods have been tested using a variety of techniques and coding systems [Hersh et al. 1998; Wilcox and Hripcsak 1998] with varying results. But the amount of work in statistical techniques applied to automated healthcare coding is still incipient. It is the authors' conviction that in this exploratory phase, the precise performance of machine learning techniques in a standalone manner must be specifically researched.

Also, in order to test the strength of machine learning techniques in languages other than English (where the majority of research has been done), the authors have done research in other languages, for instance, Spanish. This brings additional practical benefits such as the capacity for rapid processing of medical information expressed in non-English languages, from which the problems of minority healthcare disparities might be effectively mitigated. Studies show that patients not fluent in English are more likely to receive less than optimal heathcare. [Baker et al. 1998; Chalabian and Dunnington 1997; Kandula et al. 2007]. The use of interpreters and translation services of foreign medical records is expensive and error-prone. Automated coding of non-English medical records could help to bridge the gap making medical data processing less susceptible to cultural and language biases. An annotated bibliography on language barriers in healthcare settings can be found in Jacobs et al. [2003]. Likewise, the possibility of using data and tools from English-speaking countries by its non-English counterparts would assist in the diffusion of these instruments as illustrated in Johansson and Pavillon [2005].

3. BAYESIAN MACHINE LEARNING AND SHRINKAGE APPLIED TO TEXT CLASSIFICATION

This section provides the theoretical underpinnings on which our experiments are set up. We build our experimental analysis of automated ICD9-CM coding on a probabilistic framework based on the well-studied multinomial naïve Bayes classifier [Eyheramendy et al. 2003]. We consider a shrinkage-based variation of multinomial naïve Bayes introduced by [Mccallum et al. 1998] that uses the hierarchical structure of ICD9-CM to deal with the relative sparseness of training data when dealing with a large number of classes (i.e., ICD9-CM codes).

3.1 Multinomial Naïve Bayes

A multinomial naïve Bayes classifier considers that documents are generated by a probabilistic model and uses labeled training samples (sample documents preassigned to classes) to estimate the model parameters. Equipped with these estimates, the probabilistic model can be used to classify new documents by applying a Bayes theorem to determine the class with the highest probability of generating a new document. The posterior probability of each class is computed as a product of the likelihood of the data and the prior probability of the class. The new document is assigned to the class value with the maximum a posteriori (MAP) probability.

A document d is described as an ordered collection of word occurrences generated by a two-step process: (1) choose a class c_i from a list of classes $\mathcal{C}$ and (2) draw words from a vocabulary $\mathcal{V}$ using a class-specific multinomial distribution. The following considerations apply.

Class $c_j \in \mathcal{C}$ is selected with probability$\mathrm{P}(c_j)$, the prior probability of each class, using a mixture model parameterized by $\mathrm{P}(c_j)$, such that the marginal probability of generating document d is given by:

$$\mathrm{P}(d) = \sum\nolimits_{j=1}^{|\mathcal{C}|} \mathrm{P}(d|c_j) \cdot \mathrm{P}(c_j), \tag{1}$$

where

- $|\mathcal{C}|$ is the number of classes, and
- $\mathrm{P}(d|c_j)$ is the probability that document d belongs to c_j.

Word events in a document are independent and identically distributed (context and position do not matter). The position-independent probability $\mathrm{P}(w_t|c_j)$ identifies the probability of a word eventw_t in documents labeled with class c_j. $\mathrm{P}(w_t|c_j)$ parameterizes the multinomial distribution for class c_j.

Furthermore, the length in words of a document $d$is independent of the class. Document d is therefore represented as a vector of unique word counts generated from the multinomial distribution:

$$P(d|c_j) = Multinomial(n_1, n_2, .., n_m) = |d|! \ \times \ \frac{\prod_{t=1}^{m} \left[\mathrm{P}(w_t|c_j)\right]^{n_t}}{\prod_{t=1}^{m} n_t!}, \tag{2}$$

where:

- m is the number of unique words in document d;
- n_t is the number of occurrences of word w_t in document d;
- $|d|$ is the number of words in d, such that $\sum_{t=1}^{m} n_t = |d|$; and
- $\mathrm{P}(w_t|c_j)$ is the probability that word w_t occurs in documents of class c_j.

The model parameters are estimated through maximum likelihood using a set $\mathcal{D}$ of labeled document samples. The prior probabilities are estimated as the relative frequency of training document samples belonging to class c_j. Parameters $\mathrm{P}(w_t|c_j)$ are estimated as relative frequencies of word w_t in training samples belonging to classc_j.

$$\hat{P}(c_j) = \frac{\left|\mathcal{D}_j\right|}{|\mathcal{D}|} \ \hat{P}(w_t|c_j) = \frac{N_{tj}}{N_j}, \tag{3}$$

where:

–$\hat{P}\,(c_j)$ is the maximum likelihood estimate of the prior probability parameter $P(c_j)$;
–$\hat{P}\,(w_t|c_j)$ is the smoothed maximum likelihood estimate[3] of parameter $P(w_t|c_j)$;
– $|\mathcal{D}|$ is the number of training document samples;
– $\left|\mathcal{D}_j\right|$ is the number of training document samples belonging to class c_j;
– N_{tj} is the number of times w_t appears among the training documents of classc_j; and
– N_j counts the total number of words in c_j.

Given the parameter estimates $\hat{P}(c_j)$ and $\hat{P}(w_t|c_j)$ calculated in (3) and (4), respectively, any new document can be subsequently classified according to the maximum a posteriori (MAP) probability criterion. Note that the factorials in Equation (2) can be ignored as they are the same for every class and are eliminated in the normalization process. With this consideration, $P(d|c_j)$ can be expressed as the product of the

[3] Likelihood estimates over words are usually smoothed with a Dirichlet prior distribution to take care of situations where there are few or no documents in a given class. This is equivalent to adding one fictitious count to each of the frequencies and renormalizing, such that $\hat{P}(w_t|c_j) = \left(N_{tj} + 1\right)/\left(N_j + |\mathcal{V}|\right)$, where $|\mathcal{V}|$ is the number of words in the dictionary.

probabilities of each word occurrence in the document given its class. This renders the following MAP rule:

$$c_{MAP} = \underset{c_j \in \mathcal{C}}{\arg\max} \left[\hat{P}(c_j) \times \prod_{s=1}^{|d|} \hat{P}(w_s|c_j) \right], \tag{4}$$

where s is an index in the ordered sequence of word occurrences that compose document d.

3.2 Class Hierarchies and Shrinkage

Naïve Bayes classifiers rely on training samples to compute the likelihood estimates of words in classes $\hat{P}(w_t|c_j)$. For classification problems in which the number of classes is large, the lack of sufficient training samples renders parameter estimates with large variances, which in turn affect the accuracy of the classifier. If the group of classes is organized hierarchically, as in the case of ICD9-CM, the hierarchical structure can be used to compute better parameter estimates. In this work, we adapted the algorithm introduced by Mccallum [Mccallum and Nigam 1999; Mccallum et al. 1998], who used the notion of the "shrinkage" estimator, a puzzling form of smoothing first discovered by Stein [1956]. Simply stated, given several (three or more) different quantities of interest, their estimates can be improved by combining them towards a common estimate, that shrinks the mean-squared error (MSE)[4] of each individual estimator. This "shrinkage" estimator thereby constitutes a more accurate estimator than any of the individual estimators[5]. The hierarchical shrinkage algorithm described in Mccallum et al. [1998] uses training samples across a hierarchy of classes to improve each estimate $\hat{P}(w_t|c_j)$by linearly combining estimates for class levels along the class hierarchy. If $\mathcal{C} = \{c_j\}$ is the set of leaf-level classes, the "shrunk" likelihood estimate of each word w_t in documents of class c_j is calculated as:

$$\hat{P}_{sh}(w_t|c_j) = \lambda_j^{(0)} \cdot \left(1/|\mathcal{V}|\right) + \sum\nolimits_{k=1}^{r} \lambda_j^{(k)} \cdot \hat{P}(w_t|c_j^{(k)}), \tag{5}$$

where:

- r is the number of levels in the hierarchy of class c_j;
- k is the class hierarchy level, starting with 1 at the root and moving down to the leaf-level (r);
- $c_j^{(k)}$ is the k-level class along the hierarchy path of class c_j;
- $\hat{P}(w_t|c_j^{(k)})$ is the maximum likelihood estimate of w_t in k-level class $c_j^{(k)}$,calculated as the probability over words (see Equation (3)). The training data includes training samples belonging to all classes that are successors of $c_j^{(k)}$, except those belonging to class c_j;
- (0) is a dummy level added to avoid smoothing at each level of the hierarchy;
- $\lambda_j^{(0)}, \lambda_j^{(1)}, .., \lambda_j^{(r)}$ are the mixing weights for each level in the hierarchy path of class c_j, such that $\sum_{k=0}^{r} \lambda_j^{(k)} = 1$; and
- $\left(1/|\mathcal{V}|\right)$ is the uniform distribution estimate for all words w_t (a safe but very coarse estimate).

[4] This is the most commonly used risk function for an estimator in statistical decision theory.

[5] What is specially startling and counterintuitive in Stein's discovery is that the effect is valid even if the quantities are completely unrelated. For a more detailed discussion on the Stein estimator see Lee [2004].

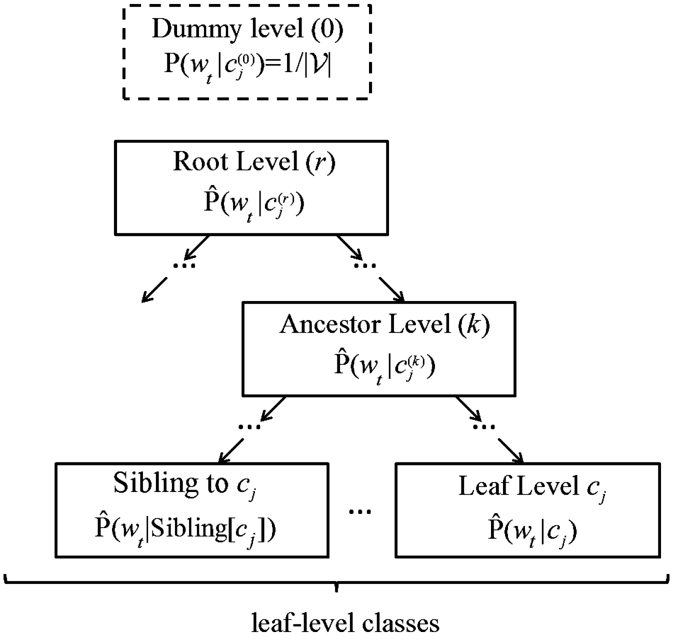


Fig. 2. A branch of a class hierarchy.

The algorithm starts by holding some portion $\mathcal{H}_j$ of the training data $\mathcal{D}$ for class c_j and uses $(\mathcal{D} - \mathcal{H}_j)$ to compute the estimates $\hat{P}(w_t|c_j^{(k)})$ at each hierarchy level. It then initializes the weights $\lambda_j^{(k)}$ to any normalized value. During the Estimation step, the algorithm estimates for each class c_j, for each level $c_j^{(k)}$ in the class hierarchy, and for each word w_t in $\mathcal{H}_j$, the probability that w_t was generated by $\hat{P}(w_t|c_j^{(k)})$. Then, during the Maximization step, the mixing weights are recalculated using the estimates calculated during the Estimation step to find the values of the mixing weights that maximize the total likelihood across all words. The procedure iterates until convergence. Figure 2 displays a tree-like representation of the class hierarchy. The full algorithm is depicted in Figure 3.

In the preceding algorithm:

- the $\beta_j^{(k)}$ values are average probability measures over all words $w_t \in \mathcal{H}_j$ representing the overall degree to which the current maximum likelihood estimates $\hat{P}(w_t|c_j^{(k)})$ predict the set of words in holdout $\mathcal{H}_j$.
- The $\lambda_j^{(k)}$ weights are recalculated at each iteration by adding and normalizing the $\beta_j^{(k)}$ values.

3.3 Related Work

There are a number of approaches recently described in the literature that are competitive in classification performance, some of which are designed to take advantage of hierarchies in document classification. Koller and Sahami [1997] used a probabilistic step refinement algorithm, initially classifying documents at a coarser level and successively refining the specificity level. Sriharsa and Avesani [2005] have described a hierarchical Dirichlet generative model for unsupervised classification of

```
EXTRACT holdout $\mathcal{H}_j$ from training dataset $\mathcal{D}$
CALCULATE the maximum likelihood estimates $\hat{\mathrm{P}}(w_t \mid c_j^{(k)})$ using training data $(\mathcal{D} - \mathcal{H}_j)$
INITIALIZE each $\lambda_j^{(k)}$ with a normalized, non-zero value such that $\sum \lambda_j^{(k)} = 1$
REPEAT
  CALCULATE each $\beta_j^{(k)} = \sum_{w_t \in \mathcal{H}_j} \frac{\lambda_j^{(k)} \cdot \hat{\mathrm{P}}(w_t \mid c_j^{(k)})}{\sum_{k=0}^{r} \lambda_j^{(k)} \cdot \hat{\mathrm{P}}(w_t \mid c_j^{(k)})}$ , $k = 0,..,r$ b // Expectation step
  READJUST each weight $\lambda_j^{(k)} = \frac{\beta_j^{(k)}}{\sum_{k=0}^{r} \beta_j^{(k)}}$ , $k = 0,..,r$ // Maximization step
UNTIL the calculation of the $\lambda_j^{(k)}$ weights converge
RETURN the $\lambda_j^{(k)}$ weights
```

Fig. 3. EM Algorithm used to determine mixing weights for each leaf-level class c_j.

text documents into a given hierarchy. Hierarchical neural networks architectures have been used by Weigend et al. [1999] and by Ruiz and Srinivasan [2002]. Cai and Hofmann [2004], proposed a hierarchical classification method that generalizes SVM learning by structuring discriminant functions in a manner that mirrors the class hierarchy. Esuli et al. [2008] have developed a hierarchical variant of Adaboost, a well-known member of the family of "boosting" algorithms for machine learning. The structural SVM algorithm developed by Tsochantaridis et al. [2004] has been applied in problems ranging from supervised grammar learning to hierarchical text classification and sequence alignment. Liu et al. [2005] have tested a hierarchical variation of SVM over Yahoo! taxonomies. The list of related work in this area could be longer as text classification has gained much interest from researchers in recent years.

4. RESEARCH DESIGN AND METHODS

This section describes the experimental assessment of the shrinkage-based classifier under analysis. The purpose of these experiments is to gather empirical evidence to (a) analyze the performance of the shrinkage-based classifier when subjected to the task of automating ICD9-CM coding, (b) compare its performance with standard (flat) text classifiers that don't make use of the hierarchical structure of the ICD coding scheme, and (c) analyze the robustness of the shrinkage-based classifier when the quality of the training dataset is questionable.

4.1 Text Classification Algorithms

A shrinkage-based classifier was applied to a set of free-text discharge diagnoses previously coded according to ICD9-CM. The classifier was compared with two well-known text classification algorithms.

— Multinomial naïve Bayes (NB), the simple generative model described in Section 3.1, has demonstrated excellent performance in text classification tasks (see Joachims [1997], e.g.).

— Support vector machines (SVM), a powerful discriminative model initially proposed by Vapnik [1995], is based on the idea of classifying data into two categories by finding an optimal hyperplane (decision boundary) that is as far away from the data of both classes as possible. Once the hyperplane is determined it can be used to separate the data points into two classes based on the relative location of the data

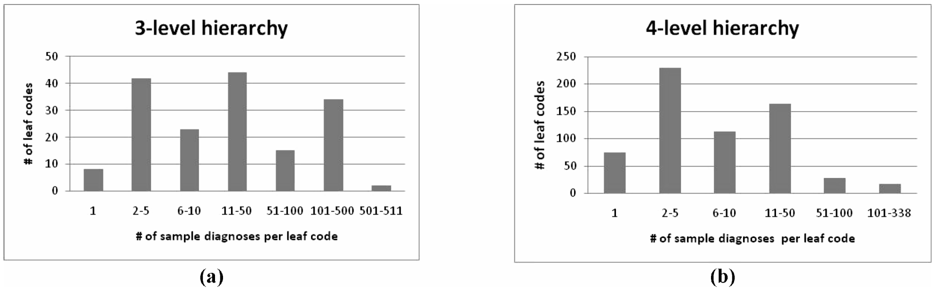


Fig. 4. Distribution of sample free-text diagnoses over leaf codes in training samples.

with respect to the hyperplane. SVM has been extended to consider multiple classes by reducing the multiclass problem to a collection of binary classification problems [Crammer and Singer 2002]. SVM implementations have been extensively tested for text classification [Cardoso-Cachopo and Oliveira 2003; Joachims 1998] and are considered state-of-the-art for their classification accuracy. For our experiments, we used a linear SVM that has computational advantages and is at the same time very efficient in classification accuracy compared to other more complex kernels.

These two classifiers are highly competitive and represent the spectrum of the most widely used algorithms for text classification both for their ease of use and classification accuracy. Both algorithms in their usual implementation can be described as "flat" classifiers since they don't consider the hierarchical nature of the text corpora being classified when this kind of structure is present in the data.

4.2 Data Source

A set of anonymized 11,776 free-text outcome diagnoses occurring in 7,380 hospitalizations was obtained. The list of discharge diagnoses was obtained from discharge abstracts in which physicians recorded this information as free-text phrases. Two experienced coders assigned corresponding codes using the Spanish Edition of ICD9-CM. For this study, original codes were aggregated at the 3rd and 4th level of the hierarchy. The 3-level class hierarchy contained a total of 408 leaf codes of which only 168 were effectively used in our dataset (i.e., these codes were effectively assigned to patients); the 4-level hierarchy included 2,687 leaf codes of which the dataset included 651.

Figure 4 graphs the distribution of sample free-text diagnoses over leaf codes for 3-level and 4-level class hierarchies for the training dataset. For the 3-level class hierarchy (Figure 4(a)), leaf codes had a mean of 63 sample diagnoses per leaf code and a median of 15, with only a small number of leaf codes tied to one sample. The 4-level class hierarchy (Figure 4(b)), was sparser as expected, with a mean of 17 sample diagnoses per leaf code and a median of 6, and a higher number of single samples per leaf code.

The documents were normalized into lower case, accents were replaced (e.g. "á" with "a"), and punctuation symbols were removed from the text. A list of common stop words in Spanish, including articles and prepositions, was also removed from the text. Not every modifier was removed from the sample documents as some of them could have semantic relevance in discriminating between diagnoses. More on this issue is covered in the discussion section.[6]

[6] The following stop words (articles, prepositions, and connectors) in Spanish were removed from the documents: el, la, los, las, a, por, según, y, o, u.

Ten percent of the document sample (1,178 documents) was randomly selected to be used as the test holdout. The remaining 10,598 documents were used to train the text classifiers. The test dataset was analyzed to check that both the vocabulary and the classes were well represented in the training dataset. We verified that 79.7% of the words in the test dataset vocabulary were present in the training dataset vocabulary and that for both class hierarchies (3-level and 4-level), 97% of the classes in the test dataset were present in the training dataset.

4.3 Experiment Design

The experiment measured the effectiveness of the shrinkage-based classifier using both clean and progressively deteriorated training data.

4.3.1 Evaluation Measure. To evaluate the effectiveness of the classifiers we measured their classification accuracy. For a given set of labeled document samples $\{d_t, c_t\}_{t=1}^{T}$selected as a test holdout, the classification accuracy evaluates the percentage of correct class predictions $\psi(d_t)$ made by the classifier. In other words,

$$\text{Accuracy } (\psi(d_t)) = \frac{1}{T} \sum \left[\psi(d_t) = c_t\right], \tag{6}$$

where d_t is each sample document in the text holdout labeled with class value c_t; T is the size of the test dataset; $\psi(d_t)$ is the class predicted by the classifier; $\left[\psi(d_t) = c\right]$ is a binary expression that returns 1 if the predicate $\psi(d_t) = c$ is true, and 0 if the predicate is false.

4.3.2 Noisy Data. Text produced for human use is often noisy for computer processing [Subramaniam et al. 2009]. Noise can come from a variety of sources: typos and misspellings, abbreviations, erroneous translation in automated processing of signals (optical character recognition, automatic speech translation). In this work noise was incrementally added on the free-text diagnoses used for training purposes by simulating typographical errors randomly selected among a list of the most common errors in Spanish (common transpositions and substitutions of letters). Table I lists the simulated errors considered in our analysis. We assumed uniform distribution of these error types in the sample. We acknowledge that this list is not exhaustive and not completely accurate, but it serves the purpose of adding real-world noise to the training set of text diagnoses and establishes a standard of simulated noisy training samples used across all three classifiers for comparison purposes. An extensive characterization of sources of noisy text can be found in Agarwal et al. [2007], Subramaniam et al. [2009], and Vinciarelli [2005].

We standardized the simulated percentage of noise by considering each word in each sample diagnose as a potential target for perturbation, excluding monosyllables. Errors were limited to one per word.

We progressively deteriorated the sample of free-text diagnoses, introducing perturbations of 25%, 50%, and 75% of the words in the training dataset. The process was repeated five times to obtain five training datasets at each specified noise level (5 × 3 = 15 noisy training datasets + 1 clean training dataset).

4.3.3 Statistical Tests. During the training stage of the experiment, we trained a classifier for every combination of text classification algorithm (shrinkage, NB, and SVM), class hierarchy (3-level and 4-level), noise level (0%, 25%, 50%, 75%), and perturbation instance at a given noise level (5 instances for each 25%, 50%, and 75% noise level). This rendered a total of 96 classifiers (3 algorithms × 2 class hierarchy levels × 3 noise levels × 5 perturbation instances) + (3 algorithms × 2 class hierarchy levels × 1

Table I. Common Errors in Spanish Due to Transpositions and Substitutions

Input string	Replacement string	Input string	Replacement string
gue	ge	mb	nb
gui	gi	nn	n
sc	s	xp	s
sc	c	rr	r
ha	a	y	ll
ho	o	x	sc
ea	ia	s	c
ns	s	k	c
bv	b	v	b
ll	y	b	v
xt	s	z	s
ee	e	j	g

Input strings found in free-text diagnoses were randomly substituted with the corresponding replacement strings to introduce noise in the training data.

zero-noise level). We tested all 96 models using the test data holdout, measuring the classification accuracy in each of them.

For each algorithm (shrinkage, NB, and SVM), we averaged the accuracies of the classifiers trained with each of the 5 perturbation instances corresponding to a given noise level (25%, 50%, 75%), and computed the standard error of each mean. This was done to account for the variability in classification accuracy due to the random nature of the noise introduced in each of the perturbed datasets (we report mean values and error bars of classification accuracy for comparison)

To determine whether there was a significant difference between the mean values of the classification accuracy for shrinkage and each of the two other classifiers (NB, SVM) at a given noise level, we used the paired-samples t-test, and Wilcoxon paired-samples signed rank test. The paired-sample Student's t-test is a parametric test that assumes that the paired differences of the repeated experiments (groups of 5 classification accuracy values in this case, corresponding to shrinkage, NB, and SVM at each noise level) are independent and identically normally distributed. Although the instances of the repeated tests themselves may not be normally distributed, their differences often are.

The Wilcoxon paired-samples signed rank test is a nonparametric test that makes no assumptions on the distribution of the samples and is probably more appropriate in this case, given the small size of the repeated tests (five for each classifier at each noise level).

All statistical tests were one-sided and performed at 5% significance level. For each noise level (25%, 50%, 75%), we performed the following comparisons: (a) Shrinkage (SH) versus NB and (b) SH versus SVM. We reported the p-values for each of the paired hypotheses tests.

4.4 Experiment Results

The results of the text classification analysis are presented in Table II and Figure 5 for ICD9-CM 3-level class hierarchies and in Table III and Figure 6 for 4-level class hierarchies. The columns labeled as "t-SH-NB" and "t-SH-SVM" report the p-values of the paired t-tests used to compare shrinkage with NB, and shrinkage with SVM, respectively. The columns labeled as "Wx-SH-NB" and "Wx-SH-SVM" report the

Table II. Classification Accuracy as a Function of % of Errors in Free-Text Diagnoses for 3-Level Class Hierachies

	3- Level									
	SH		NB		SVM		t-SH-NB	Wx-SH-NB	t-SH-SVM	Wx-SH-SVM
%errors in data	Mean	SE(*)	Mean	SE(*)	Mean	SE (*)	p (**)	p (**)	p (**)	p (**)
0.00%	84.98		78.12		80.46					
25.00%	83.78	0.67	77.1	0.86	79.78	0.79	0.0001737	0.03125	0.003967	0.03125
50.00%	82.24	0.35	74.94	0.42	78.5	0.82	1.68E-006	0.03125	0.03622	0.03125
75.00%	79.62	0.79	68.36	0.7	73.26	0.82	6.90E-005	0.03125	0.01239	0.03125

(*) test sample size: 1178 (10%) (**) one sided test, significance level = 0.05

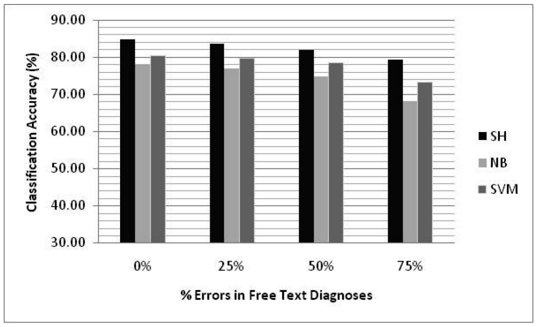


Fig. 5. Classification accuracy as a function of % of errors in free-text diagnoses for 3-level class hierarchies.

Table III. Classification Accuracy as a Function of % of Errors in Free-Text Diagnoses for 4-Level Class Hierarchies

	4- Level									
	SH		NB		SVM		t-SH-NB	Wx-SH-NB	t-SH-SVM	Wx-SH-SVM
%errors in data	Mean	SE (*)	Mean	SE (*)	Mean	SE (*)	p (**)	p (**)	p (**)	p (**)
0.00%	81.96		67.42		75.7					
25.00%	80.1	0.52	66.16	0.62	74.88	0.92	1.85E-005	0.03125	0.002509	0.03125
50.00%	78.56	0.78	62.2	0.57	70.5	0.64	3.32E-006	0.03125	0.001218	0.03125
75.00%	75.06	0.61	50.76	0.411	60.94	1.12	1.03E-006	0.03125	0.000469	0.03125

(*) test sample size: 1178 (10%) (**) one sided test, significance level = 0.05

p-values of the paired Wilcoxon rank sign tests used to compare shrinkage with NB and shrinkage with SVM, respectively.

The shrinkage algorithm outperformed both NB and SVM for all combinations of clean and noisy datasets and for both types of ICD9-CM class hierarchies (3-level and 4-level). The results of the paired t-test and the Wilcoxon paired-samples signed rank test displayed in Tables II and III show that at 5% significance, shrinkage's accuracy was higher compared to SVM and NB for all tested noise levels, and the difference in accuracy seemed to increase as the training samples were incrementally deteriorated.

In the case of the clean datasets, for 3-level hierarchies (172 class classes), shrinkage's classification accuracy reached 84.98%, while NB's was 78.12%, and SVM reached 80.46%. For 4-level hierarchies (651 classes), shrinkage's predictions were accurate 81.96% of the time, while SVM made correct predictions in 75.07% of the cases, and NB was correct 67.42% of the time.

It is interesting to note that all three algorithms have proven to be remarkably robust when subjected to training data with an increasing amount of errors, and this was

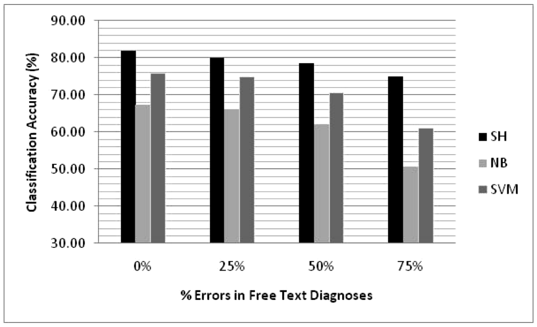


Fig. 6. Classification accuracy as a function of % of errors in free-text diagnoses for 4-level class hierarchies.

Table IV. Increase in Vocabulary Size as a Function of Noise

	0% noise	25% noise		50% noise		75% noise	
Vocabulary	n	n	%Incr.	n	%Incr.	n	%Incr.
Size	3040	4218	38.75	4530	49.01	4580	50.66

especially notable in the case of shrinkage, for which the impact of noisy training data for both 3-level and 4-level hierarchies was minimal. For 3-level hierarchy datasets, the shrinkage-based models maintained a considerably high level of accuracy which remained practically constant (1% drop) even with training datasets containing 25% errors, and dropping 5% (down to 79.62%) for a 75% noise level . For 4-level hierarchies, shrinkage accuracy went from 81.96% to 75.06% as the data was increasingly deteriorated from 0% to 75%.

As was expected, SVM outperformed NB in terms of accuracy: for 3-level hierarchies, the test accuracies (noise at 25%, 50%, 75%) were 79.78%, 78.50%, 73.26% for SVM and 77.10%, 74.94%, 68.36% for NB; for 4-level hierarchies, SVM yielded test accuracies of 25%, 50%, and 75% noise of 74.88%, 70.50%, 60.94% whereas NB yielded 66.16%, 62.20%, 50.76%.

5. DISCUSSION

Our results show that automated medical coding through statistical text classification methods is feasible. In particular, the shrinkage-based classifier is a useful tool for automated ICD9-CM coding, outperforming flat state-of-the-art text classifiers such as multinomial naive Bayes and SVM.

We were surprised by the fact that even with 50% noise there was little or no drop in accuracy, and at 75%, the degradation in accuracy was graceful (close to 7% drop). We had expected that an incremental increase in noise in the training datasets would render poorer results as the added errors should theoretically increase the size of the vocabulary and therefore reduce the frequency of features (words) in the training samples. We ran aggregated queries on the training dataset (see Table IV) and verified that the vocabulary increased from 3,040 words to a mean of 4,580 words at 75% noise (a 50% increase).

We investigated the 10 most relevant words in the clean dataset measured by discharge diagnose frequency, the number of discharge diagnoses in which a word occurs (see Yang and Pedersen [1997]). We compared this list with the most relevant words in the noisy data sets (see Table V).

We found that there was very little difference between the clean dataset and the 25%-noise dataset: Almost all of the top-10 words in the clean dataset were also present in the 25%-noise dataset. Even at 50% noise, there was plenty of commonality

Table V. 10 Most Relevant Words Measured by Discharge Diagnose Frequency

0% noise		25% noise		50% noise		75% noise	
Word	freq (%)	Word	freq (%)	Word	freq (%)	Word	freq (%)
alumbramiento	3.20	Alumbramiento	2.46	alumbramiento	1.56	parto	0.74
parto	2.99	parto	2.31	parto	1.39	alumbramiento	0.70
aguda	2.67	aguda	1.98	aguda	1.37		
hernia	2.67	hernia	1.96	hernia	1.32		
insuficienda	2.15	insuficienda	1.56	insuficienda	1.14		
litiasis	2.00	litiasis	1.42	litiasis	1.11		
embarazo	1.79	embarazo	1.32				
infeccion	1.78	infeccion	1.32				
neumonia	1.66	cesarea	1.20				
fractura	1.65						

freq(%). Diagnose frequency as a percentage over the total number of diagnoses in the data set

Table VI. Average Frequency of Words Per Class at Each Level in the ICD9-CM Code Hierarchy

	Noise %			
Level 1	0%	25%	50%	75%
Clean Words	8534	6608	4681	2754
Noisy Words	0	1926	3853	5780
Level 2	0%	25%	50%	75%
Clean Words	731	566	401	236
Noisy Words		165	330	495
Level 3	0%	25%	50%	75%
Clean Words	148	115	81	48
Noisy Words	0	33	67	100
Level 4	0%	25%	50%	75%
Clean Words	38	30	21	12
Noisy Words	0	8	17	26

in the list of top-10 words: 6 out of the top-10 words in the clean dataset were also present in the 60%-noise dataset. At 75 % noise, we found some of the relevant words still occurring, though the incidence of erroneous words became much more relevant (e.g. the typo "alunbramiento" appeared at the top of the list). This robustness exhibited by the classifier could therefore mean that, even at high levels of noise, there are enough patterns to learn in the training data. This leads us to infer that the abundance of relevant words is critical for robust automatic medical coding where the average number of words per discharge diagnose after removing all stop words is usually small (4 words per diagnose on the average for our dataset) . A large training set of medical diagnoses containing multiple occurrences of relevant words can compensate for less than optimal quality of the data rendering robust text classification models.

We also explored the average frequency of words per class at the different hierarchy levels (1 to 4). See Table VI.

As was expected, the number of "clean" words at the higher levels in the hierarchy is much larger, which improves the classification performance by providing more reliable estimates even with high levels of noise. The smoothing process provided by the shrinkage algorithm (which enhances parameter estimates for leaf-level categories with parameter estimates for their ancestors in the class) becomes critical when the frequency of words in the training samples is reduced by noise in text.

Agarwal et al. [2007] showed similar results for classifier robustness when testing flat classifiers (SVM and NB) trained with noisy datasets, but no study had been performed on noisy datasets with a very large number of classes used for hierarchical text classification as in the case of automated ICD9-CM coding.

From these results we can infer that the robustness of the shrinkage-based classifier considered in this analysis may have a direct impact on the cost incurred in producing training datasets: Predictive accuracy can be maximized with minimum data quality enhancement cost. This kind of research could help derive policy associated with data quality procedures that precede automated coding. Investing in text classification tools should help enhance automated ICD9-CM coding while maintaining low operational costs.

There are a number of issues, though, that deserve further consideration. Additional analysis may be required to better understand the nature and statistical distribution of the errors in the sample. In our experiments, we simulated the level of noise in the sample by assuming that each word in each sample diagnose had an equal chance of carrying a typographical error, and then introduced a random error selected from a list of common errors in Spanish. This had the effect of combining two distributions taken as independent events. Furthermore, we assumed that a word could not carry more than one typographical error. Although the specified procedure is valid to define a standard for comparison among classifiers, it may not accurately resemble the way in which errors appear in a real dataset. A more in-depth examination could help neutralize any bias that this procedure may have introduced.

In this work we have focused on training data with noisy features (free-text diagnoses) assuming that that discharge diagnoses used for training purposes were correctly coded. This is a common postulate in classification tasks: assuming that labels in training data are sacrosanct. But codes are assigned by human experts who manually review cases and there are multiple factors that can give way to errors of judgment, including the amount of time dedicated to review each case, the resources at hand, the training and expertise of the coders, and the complexity of the coding process. The training dataset could therefore contain noisy labels, a matter that could have an effect on the predictive accuracy of the classifier and that should be considered for further analysis.

This research restricted the analysis to 3- and 4-level hierarchies instead of the full 5-digit-level coding used by ICD9-CM. Our intent in this exploratory setting was to analyze the possible limitations of the shrinkage text classifier, as well as considering the impact on classification accuracy when trained with noisy data.

The setting of a significant probability threshold for confidently assigning a code also remains a problem. The manner in which the first appearance of a low prevalence diagnosis should be dealt with remains a question to be settled but it might reasonably be predicted that the first candidate code should render a low probability which might be resolved using thresholds. In our initial exploration, an arbitrarily set threshold lowered the correct code detection about 10%.

A qualitative analysis of the results reveals that the most frequent cause of erroneous classification, aside from coding errors in the input dataset and the incorrect handling of several low probability candidate codes, was the influence that general morphological modifiers in diagnose discharge phrases had in code assignation. The presence of indeterminate phrases such as “tumor of” or “inflammation of” (which are stripped of their complete meaning when isolated from the complete noun phrase, i.e., “tumor of the lung”) lowered the probability of codes otherwise correctly assigned by the classification algorithm. It should be noted that modifiers cannot be eliminated altogether in a rather loose fashion (we did filter a number of them as described in the Data Source section) as they may have semantic impact on the phrase and, consequently, on the code assignation.

This problem could be overcome by applying grammar-based preprocessing of the original free-text diagnoses. In order to augment the predictive power of the pure statistical approach, a dictionary of commonly appearing phrases could be constructed which, on appearing in free-text diagnoses could be assembled into “superterms” (“tumor of the lung” thus becoming “tumorofthelung”). Each superterm would consequently be treated as a single word by the classification algorithm, thus enhancing the statistically-based autocoding process. There is, of course, a trade-off with this approach, as the creation of superterms has the effect of augmenting the vocabulary and reducing word frequencies, adding variability to the maximum likelihood estimators for each word.

Assigning an ICD9-CM code to a particular set of discharge diagnoses is a complex process. As pointed by March et al. [2004], the existence of exclusions, inclusions, and secondary coding may pose NLPs with too complex requirements, and a statistical technique based on cooccurrence of words and/or phrases impresses us as a simpler approach. Nevertheless, a hybrid approach that includes some elementary form of NLP might serve to improve the rate of correct coding. Alternatively, statistical dependency models (e.g., hidden Markov models) linking modifiers and principal terms could be considered.

As we address the main goal of this article (analyzing the feasibility of automated ICD coding through a statistical classifier trained with data of varying quality), we acknowledge that we have limited our analysis to the shrinkage algorithm under study and used flat state-of-the-art classifiers as a benchmark for comparison. Our goal in this article is not that of providing a comprehensive comparison of methods. Instead, we are interested in testing automated assignation of ICD codes through statistical hierarchical text classification and training data of questionable quality using shrinkage as a proof of concept.

6. CONCLUSION AND FUTURE RESEARCH

Most of the research on text classification has revolved around classifying text documents into a flat and moderately sized set of categories. Automated ICD9-CM coding raises the bar in terms of the challenges faced by a statistical machine learning classifier: The number of class values is large (potentially very large), the amount of training data is limited, and its quality can be regarded as questionable given the nature of the medical domain and the current data collection methods in place. It is therefore important to research statistical text classification methods with the potential of overcoming these issues. Our work shows that the shrinkage-based algorithm is a valid alternative for automating the assignation of ICD9-CM even when the quality of the discharge diagnoses used for training purposes is at issue. One of the most interesting observations we made was that noise in the text seems to have limited negative impact on classification accuracy and can be effectively contained with the choice of robust statistical models such as shrinkage. As described in the discussion section, this work is not without limitations and opens a number of research avenues that deserve further exploration, among them: (a) the statistical distribution of typographical errors in discharge diagnoses; (b) the analysis of statistical learning models trained with data containing noisy labels; (c) the analysis of alternative hierarchical text classification methods, in particular hierarchical variants of SVM, given that SVM ranks at the top in terms of classification performance; (d) the feasibility of automated full 5-digit-level ICD9-CM coding with limited amounts of training data. We intend to continue our studies on the aforementioned issues and would like to encourage researchers to pursue further investigation of these approaches as a way of enhancing the automated coding process.

REFERENCES

AGARWAL, S., GODBOLE, S., PUNJANI, D., AND ROY, S. 2007. How much noise is too much: A study in automatic text classification. In *Proceedings of the 7th IEEE International Conference on Data Mining*. IEEE Computer Society.

AHIMA E-HIM^{TM} WORK GROUP ON COMPUTER-ASSISTED CODING. 2004. Delving into computer-assisted coding (practice brief). *J. AHIMA 75*, 10.

BAKER, D. W., HAYES, R., AND FORTIER, J. P. 1998. Interpreter use and satisfaction with interpersonal aspects of care for Spanish-speaking patients. *Med Care 36*, 1461–1470.

CAI, L. AND HOFMANN, T. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management.*

CARDOSO-CACHOPO, A. AND OLIVEIRA, A. L. 2003. An empirical comparison of text categorization methods. Lecture Notes in Computer Science, Springer, Berlin, 183–196.

CHALABIAN, J. AND DUNNINGTON, G. 1997. Impact of language barrier on quality of patient care, resident stress, and teaching. *Teach. Learn. Medicine: Int. J. 9*, 84–90.

CHAPMAN, W. W., CHRISTENSEN, L. M., WAGNER, M. M., HAUG, P. J., IVANOV, O., DOWLING, J. N., AND OLSZEWSKI, R. T. 2005. Classifying free-text triage chief complaints into syndromic categories with natural language processing. *Artif. Intell. Med. 33*, 31–40.

CHIERCHIA, G. AND MCCONNELL-GINET, S. 1990. *Meaning and Grammar : An Introduction to Semantics*. MIT Press, Cambridge, MA.

CHOMSKY, N. 1965. *Syntactic Structures*. Mouton, The Hague.

CHRISTENSEN, L. M., HAUG, P. J., AND FISZMAN, M. 2002. MPLUS: A probabilistic medical language understanding system. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*. Vol. 3, Association for Computational Linguistics Morristown, NJ, 29–36.

CHUTE, C. G., YANG, Y., TUTTLE, M. S., SHERERTZ, D. D., AND OLSON, N. E. 1990. A preliminary evaluation of the UMLS Metathesaurus for patient record classification. In *Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care (SCAMC)*.

COOPER, G. F. AND MILLER, R. A. 1998. An experiment comparing lexical and statistical methods for extracting MeSH terms from clinical free text. *J. Amer. Med. Inform. Assn. 5*, 62–75.

CRAMMER, K. AND SINGER, Y. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res. 2*, 265–292.

CRUSE, D. A. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge, UK.

ELKIN, P. L., CIMINO, J. J., LOWE, H. J., ARONOW, D. B., AND PAYNE, T. H. 1988. Mapping to MeSH: The art of trapping MeSH equivalence from within narrative text. In *Proceedings of the 12th Annual Symposium on Computer Applications in Medical Care (SCAMC)*. 185–190.

ESULI, A., FAGNI, T., AND SEBASTIANI, F. 2008. Boosting multi-label hierarchical text categorization. *Inf. Retr. 11*, 287–313.

EVANS, D. A., BROWNLOW, N. D., HERSH, W. R., AND CAMPBELL, E. M. 1996. Automating concept identification in the electronic medical record: An experiment in extracting dosage information. In *Proceedings of the AMIA Annual Fall Symposium*. American Medical Informatics Association, 388–392.

EYHERAMENDY, S., LEWIS, D. D., AND MADIGAN, D. 2003. On the naive Bayes model for text categorization. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*.

FISHER, C. W., LAURÍA, E., CHENGALUR-SMITH, I., AND WANG, R. Y. 2006. *Introduction to Information Quality*. MIT-IQ Press, Cambridge, MA.

FRIEDMAN, C., ALDERSON, P. O., AUSTIN, J. H., CIMINO, J. J., AND JOHNSON, S. B. 1994. A general natural-language text processor for clinical radiology. *J. Amer. Med. Inform. Assn. 1*, 161–174.

FRIEDMAN, C., SHAGINA, L., LUSSIER, Y., AND HRIPCSAK, G. 2004. Automated encoding of clinical documents based on natural language processing. *J. Amer. Med. Inform. Assn. 11*, 392–402.

GOLDSTEIN, I., ARZUMTSYAN, A., AND UZUNER, Ö. 2007. Three approaches to automatic assignment of ICD-9-CM codes to radiology reports. In *Proceedings of the Fall Symposium of the AMIA*. American Medical Informatics Association, 279–283.

HAHN, U., ROMACKER, M., AND SCHULZ, S. 2002. MEDSYNDIKATE—A natural language system for the extraction of medical information from findings reports. *Int. J. Med. Inform. 67*, 63–74.

HÉJA, G., SURJÁN, G., LUKÁCSY, G., PALLINGER, P., AND GERGELY, M. 2007. GALEN-based formal representation of ICD10. *Int. J. Med. Inform. 76*, 118–123.

HERSH, W., MAILHOT, M., ARNOTT-SMITH, C., AND LOWE, H. 2001. Selective automated indexing of findings and diagnoses in radiology reports. *J. Biomed. Inform. 34*, 262–273.

HERSH, W. R., LEEN, T. K., REHFUSS, P. S., AND MALVEAU, S. 1998. Automatic prediction of trauma registry procedure codes from emergency room dictations. *Medinfo 9* Part 1, 665–669.

HRIPCSAK, G., AUSTIN, J. H., ALDERSON, P. O., AND FRIEDMAN, C. 2002. Use of natural language processing to translate clinical information from a database of 889,921 chest radiographic reports. *Radiology 224*, 157–163.

HRIPCSAK, G., FRIEDMAN, C., ALDERSON, P. O., DUMOUCHEL, W., JOHNSON, S. B., AND CLAYTON, P. D. 1995. Unlocking clinical data from narrative reports: a study of natural language processing. *Ann. Intern. Med. 122*, 681–688.

JACOBS, E. A., AGGER-GUPTA, N., CHEN, A. H., PIOTROWSKI, A., AND HAR, E.J. 2003. Language barriers in health care settings: An annotated bibliography of the research literature. The California Endowment, Woodland Hills, CA.

JOACHIMS, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc.

JOACHIMS, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*. Springer, 137–142.

JOHANSSON, L. A. AND PAVILLON, G. 2005. IRIS: A language-independent coding system based on the NCHS system MMDS. WHO-FIC Network Meeting. WHO-FIC-2005/B.6.2.

KANDULA, N. R., LAUDERDALE, D. S., AND BAKER, D. W. 2007. Differences in self-reported health among Asians, Latinos, and Non-Hispanic Whites: The Role of Language and Nativity. *Ann. Epidem. 17*, 191–198.

KOLLER, D. AND SAHAMI, M. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning.* Morgan Kaufmann Publishers Inc., 170–178.

LAURÍA, E. AND MARCH, A. 2006. Effect of dirty data on free text discharge diagnoses used for automated ICD-9-CM coding. In *Proceedings of the 12th Americas Conference on Information Systems*.

LAURÍA, E. AND MARCH, A. 2007. Misplacing the code: An examination of data quality issues in Bayesian text classification for automated coding of medical diagnoses. In *Proceedings of the Information Resource Management Association Conference (IRMA)*.

LAURÍA, E. AND TAYI, G. K. 2003. A comparative study of data mining algorithms for network intrusion detection in the presence of poor quality data. In *Proceedings of the 8th International Conference on Information Quality (ICIQ)*. MIT, Cambridge, MA, 190–201.

LEE, P. M. 2004. *Bayesian Statistics : An Introduction*. Arnold London.

LIN, R., LENERT, L., MIDDLETON, B., AND SHIFFMAN, S. 1991. A free-text processing system to capture physical findings: Canonical Phrase Identification System (CAPIS). In *Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC)*. 843–847.

LIU, T.-Y., YANG, Y., WAN, H., ZENG, H.-J., CHEN, Z., AND MA, W.-Y. 2005. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl. 7*, 36–43.

LOWE, H. J., ANTIPOV, I., HERSH, W., SMITH, C. A., AND MAILHOT, M. 1999. Automated semantic indexing of imaging reports to support retrieval of medical images in the multimedia electronic medical record. *Methods Inf. Med. 38*, 303–307.

LUSSIER, Y. A., FRIEDMAN, C., SHAGINA, L., AND ENG, P. 1999. Automating ICD-9-CM encoding using medical language processing: A feasibility study. Res. Rep., Department of Medical Informatics, Columbia University, NY, NY.

LUSSIER, Y. A., SHAGINA, L., AND FRIEDMAN, C. 2001. Automating SNOMED coding using medical language understanding: a feasibility study. In *Proceedings of the AMIA Symposium.* American Medical Informatics Association, 418–422.

MARCH, A., LAURÍA, E., AND LANTOS, J. 2004. Automated ICD9-CM coding employing Bayesian machine learning: A preliminary exploration. In *Simposio de Informática y Salud - 33 JAII0*, Córdoba, Argentina.

MCCALLUM, A. AND NIGAM, K. 1999. Text classification by bootstrapping with keywords, EM and shrinkage. In *Proceedings of the ACL Workshop for Unsupervised Learning in Natural Language Processing.* 52–58.

MCCALLUM, A., ROSENFELD, R., MITCHELL, T., AND NG, A. Y. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the 15th International Conference on Machine Learning*.

MENDONCA, E. A., HAAS, J., SHAGINA, L., LARSON, E., AND FRIEDMAN, C. 2005. Extracting information on pneumonia in infants using natural language processing of radiology reports. *J. Biomed. Inform. 38*, 314–321.

MOORE, G. W., HUTCHINS, G. M., BOITNOTT, J. K., MILLER, R. E., AND POLACSEK, R. A. 1987. Word root translations of 45,564 autopsy reports in MeSH titles. In *Proceedings of the 11th Annual Symposium on Computer Applications in Medical Care (SCAMC)*. 128–132.

MORENO SANDOVAL, A. 1998. *Lingüística Computacional : Introducción a los Modelos Simbólicos, Estadísticos y Biológicos*. Editorial Síntesis, Madrid, Spain.

MYKOWIECKA, A., MARCINIAK, M., AND KUPSC, A. 2009. Rule-based information extraction from patients' clinical data. *J. Biomed. Inform. 42*, 923–936.

NADKARNI, P., CHEN, R., AND BRANDT, C. 2001. UMLS concept indexing for production databases: A feasibility study. *J. Amer. Med. Inform. Assn. 8*, 80–91.

RASSINOUX, A. M., MICHEL, P. A., JUGE, C., BAUD, R., AND SCHERRER, J. R. 1994. Natural language processing of medical texts within the HELIOS environment. *Comput. Methods Programs Biomed 45*, Suppl, S79–96.

RECTOR, A. L. 1999. Clinical terminology: Why is it so hard? *Methods Inf. Med. 38*, 239–252.

RUIZ, M. E. AND SRINIVASAN, P. 2002. Hierarchical text categorization using neural networks. *Inf. Retr. 5*, 87–118.

SRIHARSA VEERAMACHANENI, D. S. AND AVESANI, A. 2005. Hierarchical Dirichlet model for document classification. In *Proceedings of the ACM International Conference on Machine Learning*. 928–935.

STAUSBERG, J., LEHMANN, N., KACZMAREK, D., AND STEIN, M. 2008. Reliability of diagnoses coding with ICD-10. *Int. J. Med. Inform. 77*, 50–57.

STEIN, C. 1956. Inadmissibility of the usual estimator for the mean of a multivariate distribution. In *Proceedings of the 3rd Berkeley Symposium on Mathematics, Statistics, and Probability*. University of California Press, Berkeley, 197–206.

STRANG, N., CUCHERAT, M., AND BOISSEL, J. P. 2002. Which coding system for therapeutic information in evidence-based medicine. *Comput. Methods Programs Biomed. 68*, 73–85.

STRONG, D. M., LEE, Y. W., AND WANG, R. Y. 1997. Data quality in context. *Comm. ACM 40*, 103–110.

SUBRAMANIAM, L. V., ROY, S., FARUQUIE, T. A., AND NEGI, S. 2009. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data*.

SUOMINEN, H., GINTER, F., PYYSALO, S., AIROLA, A., PAHIKKALA, T., SALANTERÄ, S., AND SALAKOSKI, T. 2008. Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: A method description. In *Proceedings of the ICML/UAI/COLT Workshop on Machine Learning for Health-Care Applications*. M. Hauskrecht, D. Schuurmans, and C. Szepesvari Eds.

TAYI, G. K. AND BALLOU, D. P. 1998. Examining data quality. *Comm. ACM 41*, 54-57.

TREMBLAY, M., BERNDT, D., LUTHER, S., FOULIS, P., AND FRENCH, D. 2009. Identifying fall-related injuries: Text mining the electronic medical record. *Inf. Technol. Manage. 10*, 253–265.

TSOCHANTARIDIS, I., HOFMANN, T., JOACHIMS, T., AND ALTUN, Y. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st ACM International Conference on Machine Learning*.

VAPNIK, V. N. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin.

VINCIARELLI, A. 2005. Noisy text categorization. *IEEE Trans. Patt. Anal. Mach. Intell. 27*, 1882–1895.

WEIGEND, A. S., WIENER, E. D., AND PEDERSEN, J. O. 1999. Exploiting hierarchy in text categorization. *Inf. Retr. 1*, 193–216.

WILCOX, A. AND HRIPCSAK, G. 1998. Knowledge discovery and data mining to assist natural language understanding. In *Proceedings of the AMIA Annual Fall Symposium*. 835–839.

YANG, Y. AND PEDERSEN, J. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*. D. Fisher Ed. Morgan Kaufmann Publishers, 412–420.

ZELINGHER, J., RIND, D. M., CARABALLO, E., TUTTLE, M. S., OLSON, N. E., AND SAFRAN, C. 1995. Categorization of free-text problem lists: An effective method of capturing clinical data. In *Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC)*. 416–420.

ZOU, Q., CHU, W. W., MORIOKA, C., LEAZER, G. H., AND KANGARLOO, H. 2003. IndexFinder: A method of extracting key concepts from clinical texts for indexing. In *Procedings of the AMIA Symposium*. American Medical Informatics Association, 763–767.

Received October 2009; revised October 2010; accepted August 2011