Journal of Biomedical Informatics 42 (2009) 873–886

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Specializing for predicting obesity and its co-morbidities

Ira Goldstein [a,*], Özlem Uzuner [a,b]

[a] College of Computing and Information, State University of New York, University at Albany, Draper 114B, 1400 Washington Avenue, Albany, NY 12222, USA
[b] Computer Engineering Department, Middle East Technical University, Northern Cyprus Campus, Kalkanlı, Güzelyurt, KKTC, Mersin 10, Turkey

ARTICLE INFO

Article history:
Received 9 June 2008
Available online 11 November 2008

Keywords:
Classification
Combination of classifiers
Natural language processing
Machine learning

ABSTRACT

We present *specializing*, a method for combining classifiers for multi-class classification. Specializing trains one *specialist classifier* per class and utilizes each specialist to distinguish that class from all others in a one-versus-all manner. It then supplements the specialist classifiers with a *catch-all classifier* that performs multi-class classification across all classes. We refer to the resulting combined classifier as a *specializing classifier*.

We develop specializing to classify 16 diseases based on discharge summaries. For each discharge summary, we aim to predict whether each disease is present, absent, or questionable in the patient, or unmentioned in the discharge summary. We treat the classification of each disease as an independent multi-class classification task. For each disease, we develop one specialist classifier for each of the present, absent, questionable, and unmentioned classes; we supplement these specialist classifiers with a catch-all classifier that encompasses all of the classes for that disease. We evaluate specializing on each of the 16 diseases and show that it improves significantly over voting and stacking when used for multi-class classification on our data.

1. Introduction

Narrative medical records can inform many applications, including creating patient problem lists [1], assigning billing codes [2], identifying co-morbidities [3], and marking early warning signs for outbreaks of disease pandemics and epidemics [4]. However, before they can be informative for computer-supported applications, the narrative medical records must be processed with tools that can extract meaningful facts from them.

Classification provides a way of processing the content of narrative medical records. An ideal data set for tasks such as classification contains ample examples of all classes that are in question. However, data sets that include all pertinent categories, with sufficient samples from each category, are hard to obtain and even harder to create. Therefore, we are often limited to small data sets which try to represent reality. One possible characteristic of these representative data sets is the non-uniform distribution of samples among the classes. A second possible characteristic is the sparsity of the samples in some of the classes.

Although the concept of sparsity depends on the task and the representation used for the data [5], non-uniform distribution of classes and sparsity of samples can pose challenges to methods, such as classification, that are based on an exploration of statistics for representing data. In statistical classification, the small number of examples offered by a sparse, less well-represented class may adversely affect the training of a classifier on that class. Given a non-uniform distribution of classes in the data, classifiers may simply predict the well-represented classes in order to obtain high overall accuracy [6]. In binary classification it would be sufficient to have one of the classes be well represented in the data, e.g., exclusion from class A implies inclusion in class B. However, the same is not true in multi-class classification where exclusion from one class does not imply inclusion in any other specific class, e.g., exclusion from A implies inclusion in one of B, C, D... but does not specify which one.

Despite the inclination of statistical classification techniques to focus on the well-represented classes in data, the importance of the information contained in a class may not be reflected by how frequently or infrequently the class appears in the data. Even the small, sparse, less well-represented classes can contain valuable information which makes their classification worthwhile.

Our aim in this paper is to improve the classification performance on the less well-represented classes in multi-class classification of diseases based on information in medical discharge summaries. While improving performance on less well-represented classes, we also aim to maintain overall performance on the task. Given a discharge summary of a patient, our task is to predict the status of the patient with respect to obesity and 15 of its co-morbidities. We treat the prediction for each disease as an independent multi-class classification task. Each of the 16 diseases can be classified as being present, absent, or questionable in the

* Corresponding author. Fax: +1 518 442 5367.
E-mail address: ig4895@albany.edu (I. Goldstein).

1532-0464/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2008.11.001

patient, or unmentioned in the discharge summary of the patient. We observe that the data for these tasks exhibit both non-uniformity and sparsity. In other words, there is an imbalance in the distributions of present, absent, questionable, and unmentioned classes for each disease, e.g., there are orders of magnitude more examples of the unmentioned class than the questionable class for most diseases, and some classes contain a very small number of examples, e.g., the absent class contains only 152 of the 19,695 judgments across all diseases.

To improve macro-averaged performance on multi-class classification in the presence of non-uniform distribution of data and in the presence of sparse, less well-represented classes, we develop and present *specializing*. Specializing is a novel method for combining classifiers. In general, it works as follows: for a multi-class classification task, for each class in the data, specializing selects a *specialist classifier* from a set of available complementary classifiers that are trained in a one-versus-all (OVA)[1] manner to predict that class. In other words, the specialist classifier for each class focuses on distinguishing that class from the rest. Specializing then combines the specialist classifier for each class with a *catch-all classifier* that is trained as a multi-class classifier[2] for the task and can distinguish all classes for that task from one another. In terms of performance metrics, the specialist classifier for each class is the highest *F*-measure OVA-trained classifier for that class whereas the catch-all classifier gives the highest micro-averaged *F*-measure across all classes for the task. The combination of the specialist classifiers with the catch-all classifier produces the *specializing classifier* for the task.

We combine the specialist and the catch-all classifiers by allowing the specialist classifiers to predict their classes before the catch-all classifier labels those samples that fail to receive a definitive class assignment from any of the specialist classifiers. This approach avoids having to handle contradicting assignments from competing specialist classifiers by running the specialist classifiers in a strict order, starting with the specialist for the least well-represented class, working its way up to the most well-represented class. Each specialist classifier only runs on those samples that do not receive a definitive class assignment from the specialists that run before it. This strict order ensures that the specialists for the less well-represented classes are given due consideration before the specialists for the better represented classes are run. For those samples that do not receive a definitive class assignment from any of the specialist classifiers, the catch-all classifier is run at the end, and always assigns a class.

For predicting the status of patients with respect to obesity and 15 of its co-morbidities, we treat the classification of each disease as an independent multi-class classification task, i.e., we have 16 independent multi-class classification tasks and each disease can be present, absent, or questionable in the patient, or unmentioned in the discharge summary of the patient. In this paper, we refer to obesity and its 15 co-morbidities as *diseases*. We refer to present, absent, questionable, and unmentioned as *classes*. We apply specializing to each disease separately. We create specialist classifiers for each of the present, absent, questionable, and unmentioned classes of each disease. We supplement the specialist classifiers for each disease with a catch-all classifier that can distinguish present, absent, questionable, and unmentioned classes from each other for that disease. We choose the specialist and the catch-all classifiers from C4.5 decision trees, Naïve Bayes classifiers, and AdaBoosted decision stumps. By combining the specialist and catch-all classifiers for a disease, we create the specializing classifier for that disease.

[1] In a multi-class data set, a one-versus-all classifier recognizes only one class and learns to distinguish it from all others, e.g., class A versus not class A.

[2] A multi-class classifier learns to recognize multiple classes at the same time, i.e., distinguish all of classes A, B, C, D, etc., from each other.

2. Related work

Specializing trains classifiers with complementary strengths and combines these classifiers in a novel manner to create a specializing classifier per task.

2.1. Combining classifiers

In the literature, various approaches to combining classifiers have been presented. Voting, stacking, and boosting are among these approaches. Voting combines classifiers in various ways, such as selecting the class assigned by the majority of the classifiers or taking the average or the product of the probabilities that the item is correctly classified (i.e., how strongly the classifier believes in the prediction). Stacking [7] applies cross-validation to combining individual classifiers while adjusting for the biases of these classifiers. More specifically, stacking attempts to figure out the bias of each of the classifiers with respect to the training data. It then adjusts for the determined bias. Stacking can be employed either to combine different classifiers or to improve the performance of a single classifier. Boosting iteratively combines many "weak learners" from a classifier [8–10]: at each iteration, it trains a weak learner that tries to improve performance on the samples that have not been adequately learned by past weak learners.

2.2. Combined classifiers

The literature has shown that ways of combining classifiers vary in their merits. Chan and Stolfo [11–13] addressed the issue of scaling of data sets (i.e., the problems that arise as data sets grow larger than available computer memory) by dividing the training data set into subsets and by training various classifiers on these subsets. They presented *combiners* and *arbiters* as two approaches for combining classifiers. Combiners learn the relationship between the output of the individual classifiers and the correct classification in order to provide a prediction. Arbiters are classifiers which are combined with an arbitration rule, and can either arbitrate among the individual classifiers or provide their own predictions. Both combiners and arbiters are forms of conflict resolution among classifiers. Zenko et al. [14] evaluated seven approaches for combining classifiers on 21 data sets, finding that stacking outperformed boosting. Liu et al. [15] recognized that many classification methods discard all but the highest performing classifier and proposed a Combination Strategy for Multi-class Classification (CSMC). CSMC employed set theory and evidence weight, retaining multiple rules to combine into a classifier for multi-class multi-label classification. CSMC weighted rules based upon their frequency in the data, and saw improvement over individual classifiers including C4.5. Daskalakis et al. [16] used a "panel of classifiers" (i.e., statistical quadratic Bayesian, *k*-nearest neighbor, and probabilistic neural network classifiers) to develop a multi-class classifier for biopsies of thyroid nodules. Each classifier in this panel approached the classification problem from a different perspective. This "panel of classifiers" used majority voting and resulted in a significant improvement over the use of the best single classifier. Similarly, Eom et al. [17] compared individual classifiers to "ensemble models" and tested them against four data sets (i.e., cardiovascular disease, pulmonary complaints, tuberculosis, and cancer). The "ensemble models" combined individual classifiers whose errors differed from one another (i.e., there was minimal overlap in classification errors). Each of the "ensemble models" outperformed the best performing individual classifier on each of the four data sets.

2.2.1. Characteristics of combined classifiers

Duin et al. [18], when analyzing the performance of various combined classifiers, concluded that "there is no overall winning

combining rule" for classifiers. The selection of classifiers and of approaches to combining those classifiers depends upon the data, especially the number of classes to be classified. Tax et al. [19] demonstrated this by comparing multiple rules for both binary and multi-class handwritten digit recognition. Surprisingly, they observed that when dealing with a binary problem, there was no difference in the average results between combining three classifiers and combining ten classifiers (i.e., on average, the combined classifiers performed equally well on binary classification). However, when there are more than two classes (i.e., multi-class classification), differences in the results appear with various classifier combinations.

Hu and Damper [20] took a theoretical approach to extend Kuncheva's [21] "no panacea" principle which stated that when combining two classifiers in binary classification problems, there is no perfect combination of algorithms for all situations (i.e., one size does not fit all). Hu and Damper showed that the principle applies when there are more than two classifiers and for multi-class classification.

We extended the above studies by developing specializing, a method for combining statistical classifiers for multi-class classification of diseases based upon information in discharge summaries. Specializing differs from approaches in literature in its novel mixture of several performance enhancing ideas inasmuch as it: (1) takes advantage of classification algorithms that can learn samples that are hard to learn even when the attribute sets are relatively large and noisy, (2) trains these classifiers in a way, specifically in an OVA manner, that allows them to specialize on even the less well-represented classes, (3) uses the best OVA-trained classifier for a class as the specialist for that class, (4) utilizes information about each class to impose a priority order on the specialists, (5) takes advantage of the difference in the focus of multi-class classification from one-versus-all classification in order to supplement the specialist classifiers with catch-all classifiers, and (6) makes use of the complete data set rather than subsets of data in order to maintain the most accurate representation of less well-represented classes throughout. Specializing allows a single classifier to make an assignment for a specific class. It combines classifiers in a sequential manner in order to arrive at a definitive class assignment for each of the samples in the data. Sequential activation of classifiers eliminates the need for an additional conflict resolution strategy.

3. Materials and methods

The data set for the study presented in this paper was developed for the i2b2 Shared-Task and Workshop on Challenges in Natural Language Processing for Clinical Data: Obesity Challenge [3]. This data set consisted of medical discharge summaries which had been annotated by doctors for obesity and 15 of its co-morbidities. We split this data set into training and test sets for each disease. We linguistically processed the discharge summaries in the data and extracted attributes with which to represent their text (see Section 3.3 for details). In order to automate the task of labeling obesity and co-morbidities as present, absent, or questionable in the patient, or unmentioned in the discharge summary of the patient, we set up one multi-class classification task per disease. We trained J48 decision trees, Naïve Bayes, and AdaBoost.M1 classifiers using 10-fold cross-validation on the training data for each disease. For each of the 16 diseases, we designated one specialist classifier per class and a catch-all classifier per disease based on cross-validation on the training set for that disease. We created the specializing classifier for each disease from the specialist classifiers and the catch-all classifier for that disease.

3.1. Data

The data for this study consisted of 1238 discharge summaries from Partners HealthCare. This data had been fully de-identified and then annotated by obesity experts for information on obesity and its co-morbidities. Two obesity experts annotated each discharge summary and determined whether obesity and its 15 most frequent co-morbidities were present (marked with a Y in the data set), absent (marked with an N in the data set), or questionable (marked with a Q in the data set) in the patient according to explicitly stated text in the discharge summary, or unmentioned (marked with a U in the data set) in the text. In cases where the two obesity experts disagreed, the discharge summary was annotated by a third expert. Majority decision among the experts determined the final judgment on each of the 16 diseases. In the absence of a majority vote, some diseases remained without a judgment. In other words, some records contained final judgments only for a subset of the 16 diseases. The Institutional Review Boards of SUNY, Albany and Partners HealthCare approved this study.

3.2. Training and test sets

As detailed by disease in Table 1, following the division of the data released as part of the i2b2 Shared-Task, we employed 59% of the discharge summaries as our training data and set aside 41% of the discharge summaries as our test data. The training data contained 11,630 class assignments across the 16 diseases. The test data contained 8065[3] class assignments across the 16 diseases. The training data for each disease included a subset of the 59% of the discharge summaries set aside for training. The summaries that did not have a final judgment for a disease were omitted from the training data for that disease. Similarly, the test data for each disease included a subset of the 41% of the discharge summaries set aside for testing. The summaries that did not have a final judgment for a disease were omitted from the test data for that disease. Table 1 shows the non-uniform distribution of classes in the data across 16 diseases and indicates the absent and questionable classes as smaller, sparse, less well-represented classes in all diseases.

3.3. Feature extraction

Linguistic processing of narratives can expose attributes that can contribute to more accurate representation of the narratives' contents. The choice of linguistic attributes represented and utilized for a task depends on the task. These attributes can be based upon syntax, semantics, or even just surface processing. As the focus of this article is on specializing as a method for classifying diseases, we rely on only a basic set of attributes, namely stemmed lowercase words and the polarity (i.e., positive, uncertain, or negative) of the assertions made on terms corresponding to medical problems (i.e., diseases and symptoms).

In order to extract these features from discharge summaries, before classification, we syntactically processed the discharge summaries to detect terms corresponding to medical problems. We then semantically processed the discharge summaries to determine the polarity of the assertions made on the identified terms. Once the polarity was determined, we transformed the text to distinguish the negative and uncertain assertions from positive assertions. Finally, we converted the transformed text to lower case and applied stemming [22] in order to conflate variations of words.

We used a binary term vector space model [23] to represent our data. In this vector space, we represented each discharge summary

[3] Following the studies in this manusript, the i2b2 Shared-Task organizers eliminated some of the test samples from the data. As a result, the final i2b2 Shared-Task test data included only 8044 class assignments.

Table 1
Ground truth.

	Number of samples in the training data					Number of samples in the test data				
Disease	Y	N	Q	U	Total	Y	N	Q	U	Total
Asthma	93	3	2	630	728	68	2	2	433	505
Atherosclerotic CV disease	399	23	7	292	721	278	22	2	196	498
Heart failure	310	11	0	399	720	206	11	2	280	499
Depression	104	0	0	624	728	72	0	0	435	507
Diabetes mellitus	485	15	7	219	726	339	12	3	150	504
Gallstones/cholecystectomy	109	4	1	615	729	88	2	0	418	508
GERD	118	1	5	599	723	69	1	1	434	505
Gout	90	0	4	634	728	52	0	0	454	506
Hypercholesterolemia	304	13	1	408	726	213	6	4	280	503
Hypertension	537	12	0	180	729	375	6	3	121	505
Hypertriglyceridemia	18	0	0	711	729	10	0	0	498	508
Osteoarthritis	115	0	0	613	728	86	0	0	417	503
Obesity	298	4	4	424	730	198	3	3	290	494
Obstructive sleep apnea	105	1	8	614	728	69	0	2	433	504
Peripheral vascular disease	102	0	0	627	729	64	0	0	444	508
Venous insufficiency	21	0	0	707	728	10	0	0	498	508
Total	3208	87	39	8296	11,630	2197	65	22	5781	8065

Y, present; N, absent; Q, questionable; and U, unmentioned. Number of samples in each class in the training and test sets in the data from i2b2 Shared-Task and Workshop on Challenges in Natural Language Processing for Clinical Data: Obesity Challenge.

by an attribute vector. Each attribute in an attribute vector was mapped to a dimension of the vector space. We used attribute vectors as input to the classifiers in Weka [24].

3.3.1. Syntactic processing

In order to determine the polarity of medical problem assertions (see Section 3.3.2), we first identified terms corresponding to medical problems in our discharge summaries. We defined medical problems as diseases and symptoms. We based our medical problem terms upon a target list of Unified Medical Language System (UMLS) semantic types (see Table 2) and marked them using the MetaMap [25] Java API (MMTx). More specifically, we identified the noun phrases (NP) in each report and studied the possible semantic types of the NPs based on UMLS. For each NP, we checked its possible semantic types for ones that are included in the target list in Table 2. Matching the semantic type of an NP and its headword to a member of the target list concluded the process and marked the whole NP as containing a term corresponding to a medical problem. Failure to achieve a match for a noun phrase required that we search its sub-phrases and the headwords of the sub-phrases for UMLS semantic types included in the target list. The sub-phrase that matched one of the target list semantic types, and whose headword also matched one of those semantic types, is marked as a term corresponding to a medical problem.

Table 2
Target list of UMLS semantic types used to match medical problems.

Disease		Symptom	
Code	Description	Code	Description
acab	Acquired abnormality	clan	Clinical attribute
anab	Anatomical abnormality	diap	Diagnostic procedure
bact	Bacterium	fndg	Finding
cgab	Congenital abnormality	lbpr	Laboratory procedure
comd	Cell or molecular dysfunction	lbtr	Laboratory or test result
dsyn	Disease or syndrome	sosy	Sign or symptom
inpo	Injury or poisoning		
mobd	Mental or behavioral dysfunction		
neop	Neoplastic process		
patf	Pathologic function		
virs	Virus		

UMLS semantic types corresponding to medical problems, i.e., diseases and symptoms.

3.3.2. Semantic processing

Physicians often assert uncertain or negative diagnoses in narrative medical records [26]; for example, to provide information that contrasts with the positive diagnoses [27] or to keep track of all potential diagnoses that have been considered. Unless correctly identified, the presence of negative and uncertain assertions in the narrative of medical records can be confused with positive assertions and adversely affect automated system performance. Several research efforts focused on identifying and making use of uncertain and negative assertions in text. Mutalik et al. [28] showed that the UMLS can be used to reliably detect negated concepts in medical narratives. Sibanda [29] extended NegEx [30], by taking a rule-based approach, to identify not just positive, negative, and uncertain assertions, but also assertions made in reference to someone other than the patient. Named Extended NegEx, this system was developed on medical discharge summaries.

We employed the source code of Extended NegEx [29] to study the nature of the assertions in discharge summaries. We applied Extended NegEx to terms corresponding to medical problems, as identified in Section 3.3.1. In order to distinguish the positive, negative, and uncertain assertions from one another in the attribute vector, we left the positive assertions unchanged, but transformed the negative and uncertain assertions in the following manner: assertions that were identified as negative were repeated within the narrative, but the medical problem term was pre-pended with "abs" (e.g., "Patient denies fever" becomes "Patient denies fever absfever"); assertions that were identified as uncertain were repeated in the narrative, but the repetition included the medical problem term pre-pended with "poss" (e.g., "possible pneumonia" becomes "possible pneumonia posspneumonia"). We call these transformed terms *asserted medical problems*.

3.3.3. Surface processing

Our last data preparation step involved morphological analysis. In order to ensure that morphologically similar words were brought together as a single attribute, we converted the text transformed via semantic processing to lower case and applied stemming [22] (e.g., both "CHRONIC" and "chronically" were converted to "chronic").

3.4. Weka

Stemmed lowercase words and asserted medical problems provide a good start to predicting whether a disease is present, absent,

or questionable in the patient, or unmentioned in the discharge summary of the patient. However, each discharge summary can make multiple references to a medical problem. The polarity of the assertions made on each mention of the medical problem can be different, requiring us to further process these attributes to determine the most likely meaning of their combination and the implications for the presence of a disease in a patient based on that discharge summary. Classification provides a means for further processing these attributes.

We performed classification using Weka [24], an open source collection of machine learning algorithms. We employed Weka version 3.5.5's classification algorithms with stemmed lowercase words (i.e., we discarded any non-alphabetic content) and asserted medical problems. For each of our 16 diseases, we identified the stemmed lowercase words and asserted medical problems relevant for classifying a disease from the training data for that disease. In order to eliminate attribute selection as a factor in system performance, we chose to use all of the thus identified attributes, including stop words, for a disease in classifying that disease. We treated all attributes for a disease in the same manner, i.e., we did not differentiate between the stemmed lowercase words and the asserted medical problems. The number of attributes used for each disease was more than 2400.

We employed the identified attributes for each disease to build a term vector space model for data representation for that disease. Each identified attribute matched to a dimension of the term vector space. The dimensions of the term vector space determined the dimensions of attribute vectors of medical discharge summaries. The attribute vector of each discharge summary noted the presence or absence of each attribute within that medical discharge summary. This gave equal weight to all of the attributes regardless of the number of times each attribute appeared within the given medical discharge summary.

We trained and 10-fold cross-validated the classifiers for each disease on the training set and therefore on the term vector space for that disease. We evaluated the classifiers for each disease on the test set for that disease, using the term vector space created from the training set for that disease.

3.5. Specializing

Specializing aims to improve classification of less well-represented classes while maintaining overall performance in multi-class classification. It takes three major steps to achieve this goal: (1) for each class in a classification task, specializing trains complementary classifiers in an OVA manner and forces them to learn to

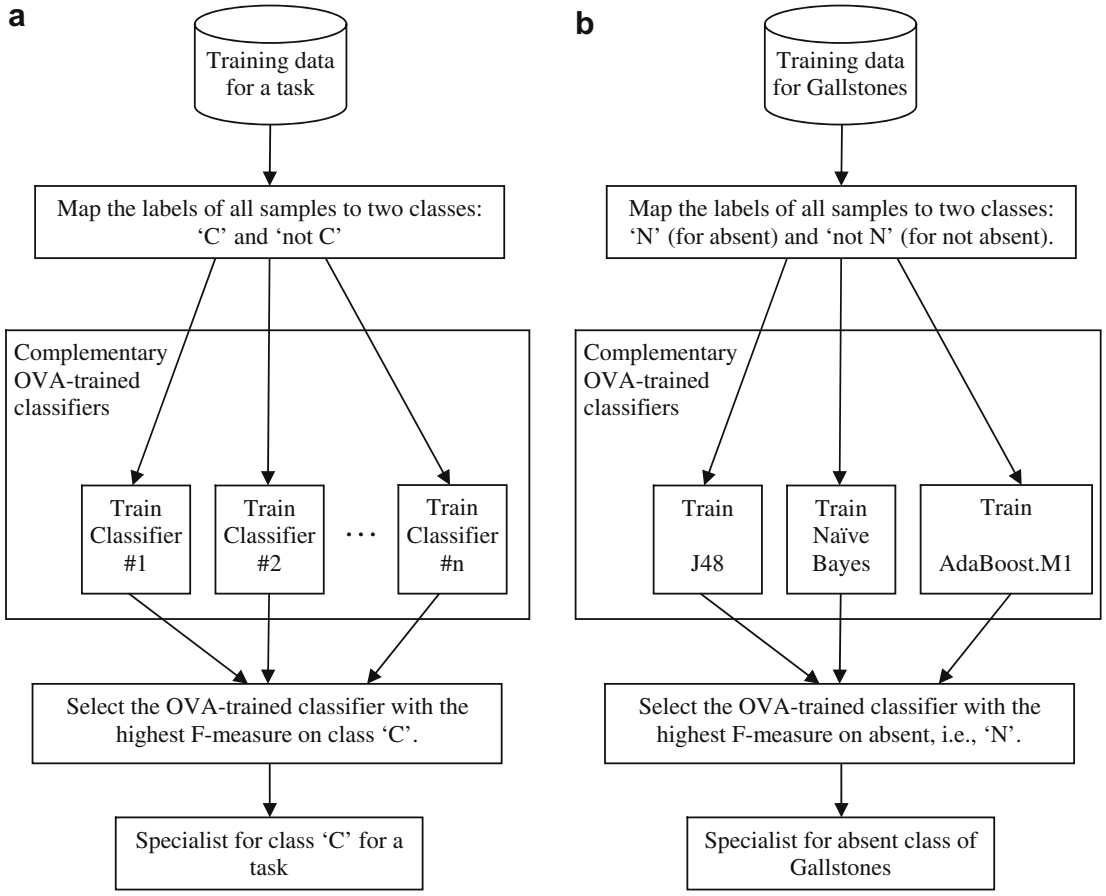

Fig. 1. General and applied processes for selecting specialist classifiers. (a) shows the general process for selecting a specialist classifier. This process begins by mapping the labels of all samples to two classes. The two classes consist of those samples that are in a given class which we refer to as 'C', and all samples not in the given class which we refer to as 'not C'. The specialist for class 'C' is selected from a set of complementary classifiers trained in a one-versus-all (OVA) manner on class 'C' and has the highest non-zero *F*-measure on class 'C'. (b) demonstrates the selection of a specialist for the absent class, class 'N', of gallstones. For gallstones, class 'N' consists of those records where gallstones is absent in the patient according to explicitly stated text in the narrative of the discharge summary. The class 'not N' consists of those records where gallstones is either present (Y) or questionable (Q) in the patient, or unmentioned (U) in the discharge summary of the patient. Class 'not N' is created by consolidating classes Y, Q, and U and transforming their labels to 'not N'. The specialist for class 'N' of gallstones is selected from among C4.5 decision trees (J48), Naïve Bayes, and AdaBoost.M1 classifiers trained in an OVA manner to separate class 'N' from 'not N'. The OVA-trained classifier with the highest non-zero *F*-measure is selected as the specialist classifier for class 'N' of gallstones.

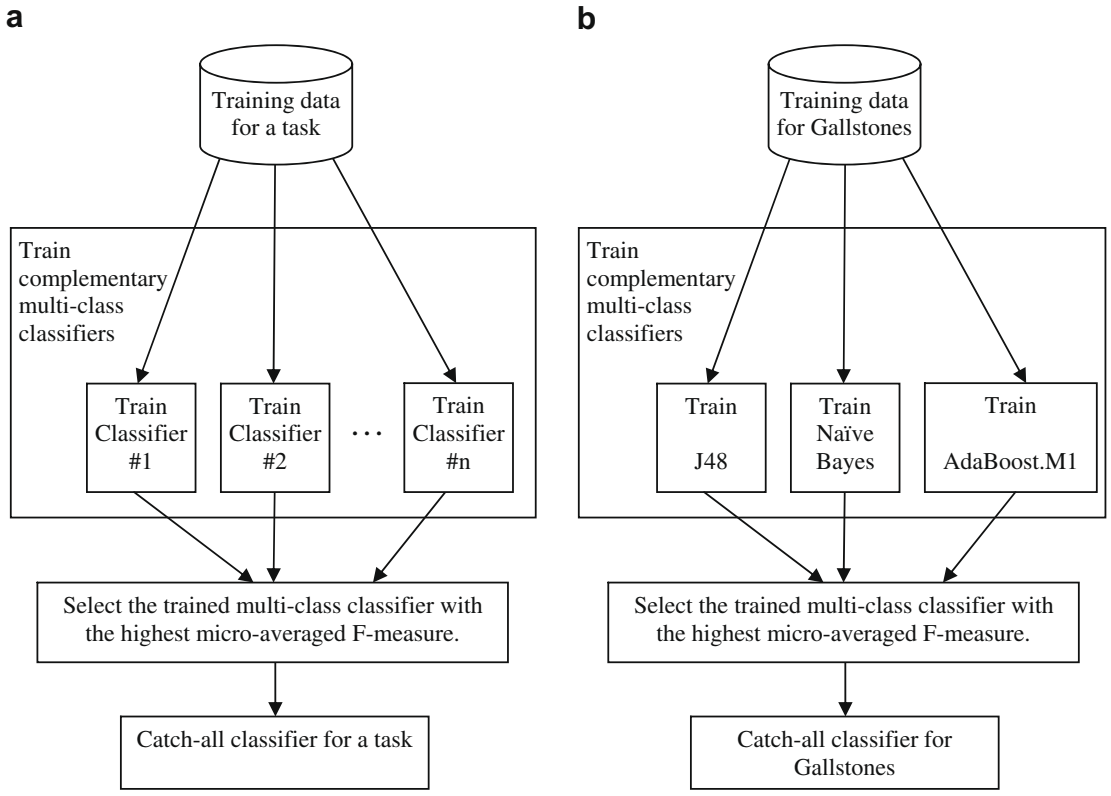

Fig. 2. General and applied processes for selecting catch-all classifiers. (a) demonstrates the general process for selecting a catch-all classifier for a task. This process begins by training complementary multi-class classifiers that can distinguish all classes for a task from one another. The multi-class classifier with the highest micro-averaged *F*-measure across all classes is selected as the catch-all classifier for the task. (b) applies the process in Fig. 2a to selecting a catch-all classifier for the disease gallstones. We find that the disease can be present (Y), absent (N), or questionable (Q) in the patient, or unmentioned (U) in the discharge summary of the patient. The catch-all classifier for gallstones is selected from among C4.5 decision trees (J48), Naïve Bayes, and AdaBoost.M1 classifiers trained in a multi-class manner to distinguish Y, N, Q, and U classes of gallstones from one another. It is the single multi-class classifier with the highest micro-averaged *F*-measure across all classes for gallstones.

distinguish the items in that class from all of the items in all of the other classes [31]. It selects a specialist classifier from these OVA-trained classifiers (see Fig. 1a). (2) Specializing supplements the specialist classifiers with a catch-all classifier per classification task. The catch-all classifier for a task is a multi-class classifier and aims to capture the distribution of samples in all of the classes in the task (see Fig. 2a). (3) Specializing combines the specialists classifiers and the catch-all classifier for the task in a manner that ensures due consideration to all of the classes in the data (see Fig. 3a).

For classifying obesity and 15 of its co-morbidities, we treat each disease independently of the others and apply specializing to each disease separately. We treat the classification of each disease as a multi-class classification task in which the disease is classified as being present, absent, or questionable in the patient, or unmentioned in the discharge summary of the patient. For each of the present, absent, questionable, and unmentioned classes of a disease, we train J48, Naïve Bayes, and AdaBoost.M1 classifiers in an OVA manner and select from them a specialist per class (see Fig. 1b). For each disease, we train J48, Naïve Bayes, and AdaBoost.M1 classifiers in multi-class manner and select from the trained classifiers a catch-all classifier for the disease (see Fig. 2b). The specializing classifier for each disease combines the specialist classifiers for the classes of that disease with the catch-all classifier for that disease (see Fig. 3b).

3.5.1. Complementary classifiers

Specializing relies upon classifiers with complementary strengths. Each of the employed classifiers can be an individual classifier or a combined classifier. Specializing treats each classifier as a "black box", selecting a trained classifier based solely upon its *F*-measure (see Section 3.6.1).

For predicting the status of patients with respect to obesity and 15 of its co-morbidities, we based our specializing classifier on three commonly used multi-class classifiers: Naïve Bayes, C4.5, and AdaBoost.M1. The choice of these classifiers was motivated by their complementary strengths given the focus on learning less well-represented classes. Naïve Bayes classifiers have been shown to be effective when the number of attributes exceeds the number of observations [32]. However, given the possible noise in these attributes, we complemented Naïve Bayes with a widely used C4.5 classifier (Weka's J48) which tends to be robust to noisy data [33]. Finally, we added AdaBoost.M1 [34], a multi-class implementation of boosting for combining many "weak learners" to learn the samples that are harder to classify [8–10].

3.5.1.1. C4.5. (J48) decision tree. Weka's J48 decision tree classifier is an implementation of C4.5, which is a greedy, divide and conquer, statistical learning algorithm. J48 decision trees use the attributes of the data to recursively split the data into smaller subgroups [35,36]. At each step, the best split, as determined by the gain ratio [35], is selected from all possible splits across all attributes. Gain ratio is based upon both information gain and split information. Information gain measures the change in information entropy [37] due to a given split. Split information measures the level of homogeneity (in terms of class distributions) of data in a split. Gain ratio normalizes information gain by the split information. We included J48 in our specializing classifier due to its robustness to noisy data.

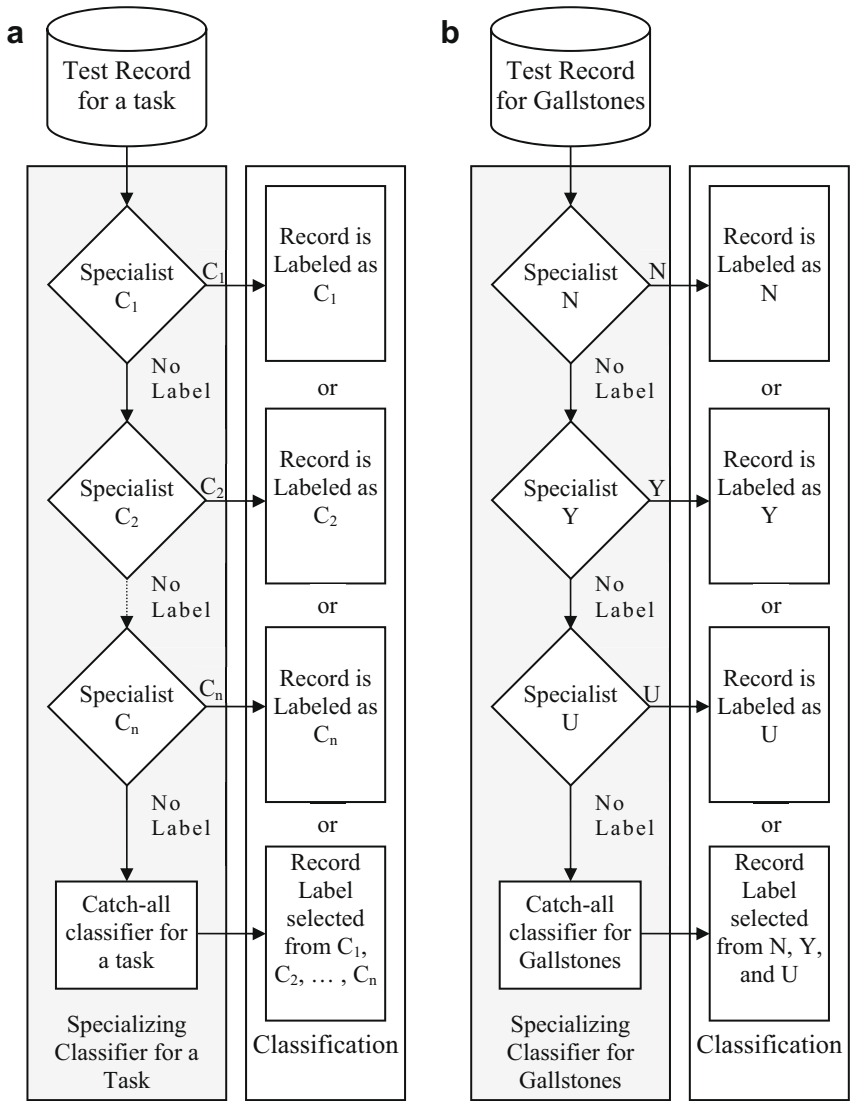

Fig. 3. Specializing classifiers for a general task and for the disease gallstones. (a) shows that the specializing classifier for a general task consists of one specialist classifier per class (C_1 through C_n) and a catch-all classifier for the task. Each specialist classifier has an F-measure >0 and either predicts the class it specializes on or fails to make any prediction. The specialist classifiers are combined in a sequential manner starting with the specialist for the least well-represented class, working up to the most well-represented class. For each data sample, the process continues until the sample is labeled or until all specialist classifiers have attempted and failed to classify the sample. The specialist classifiers are then supplemented with a multi-class catch-all classifier. The catch-all classifier ensures that a class assignment is made to each sample. (b) shows the specializing classifier for the disease gallstones. For this disease, the sample sizes for the classes in the training data are Y = 109, N = 4, Q = 1, and U = 615. Specializing classifier for gallstones combines specialists in order N, Y, and U, going from the least well-represented to the most well-represented class. The specialist for Q for gallstones had an *F*-measure = 0; therefore, it is omitted from the specializing classifier. Starting with the specialist classifier for class 'N', given a data sample, either the specialist labels the sample with an 'N' and the process stops, or the process moves on to the next specialist classifier. The specialist classifiers are supplemented by the catch-all classifier for gallstones. The catch-all classifier for gallstones was trained on the same data used to train the specialist classifiers for this disease and for gallstones it can assign a Y, N, Q, or U to the sample.

3.5.1.2. Naïve Bayes. Naïve Bayes classifiers apply Bayes' theorem, assuming a naïve (i.e., strong) independence among the attributes, i.e., that the conditional probability of an attribute given a class is independent of other attributes given the class. Naïve Bayes uses joint probabilities to estimate the likelihood that an item belongs to a specific class [38]. Naïve Bayes has been shown to be effective when the number of attributes exceed the number of items in the training set [32].

3.5.1.3. AdaBoost.M1. Boosting improves the accuracy of a classifier by iteratively training classifiers on subsets of the training data [39]. Among these classifiers, it retains those that perform marginally better than chance. The retained classifiers are combined and often show improved performance. We employed the multi-class Adaptive Boosting (AdaBoost.M1) implementation of this algorithm. AdaBoost.M1 weights the results from many binary decision stumps to build a multi-class classifier whose results typically improve over any of the decision stumps. AdaBoost.M1 complements J48 and Naïve Bayes due to its ability to learn the samples that are harder to classify.

3.5.2. Specialist classifiers

Specializing identifies a specialist classifier per class by training complementary classifiers in an OVA manner. For each class, it focuses only on those OVA-trained classifiers that could correctly assign samples to that class. It selects from these successfully trained OVA classifiers the one with the highest *F*-measure on that class (see Section 3.6.1 for a discussion on *F*-measure) as the specialist classifier for that class (see Fig. 1a). The specialist for a class can only make a binary decision and either assigns that class or fails

to assign any class, i.e., the specialist for class A can predict either A or not A, where an assignment of A implies a definitive assignment of A but an assignment of not A implies that the definitive class is not known by the specialist for class A.

For the task presented in this paper, the specialist classifiers are chosen from OVA-trained J48, Naïve Bayes, and AdaBoost.M1 classifiers (see Fig. 1b). On our training data, the *F*-measures of the specialist classifiers for all classes of all diseases had a mean of .82 and a standard deviation of .10.

3.5.3. Catch-all classifiers

Catch-all classifiers supplement specialist classifiers. The catch-all classifier for a task is the multi-class classifier with the highest micro-averaged *F*-measure across all of the classes in that task. Unlike the specialist classifiers, the catch-all classifier can assign any of the classes in the task (see Fig. 2a). For the task in this paper, the catch-all classifiers are chosen from multi-class-trained J48, Naïve Bayes, and AdaBoost.M1 classifiers (Fig. 2b). On our training data, the *F*-measures of the catch-all classifiers for all diseases had a mean of .91 and a standard deviation of .06.

3.5.4. Specializing classifiers

Specializing combines the specialist and catch-all classifiers for a classification task by employing these classifiers in a strict order. Given already trained specialist and catch-all classifiers for a task, to classify an unknown data item, specializing first calls the specialist classifiers in increasing order of class size (see Fig. 3a). It starts by invoking the specialist classifier for the least well-represented class in the data. This specialist attempts its classification before the specialist classifier for the second least well-represented class is invoked. The specialist for the second least well-represented class is invoked only if the item fails to receive a definitive class assignment from the specialist classifier for the least well-represented class. The specialist for the second least well-represented class is followed by the specialist for the third least well-represented class, but only if the item fails to receive a definitive class assignment from the specialist for the second least well-represented class, etc. This process continues until the item receives a definitive class assignment or until all of the specialist classifiers have attempted and failed to classify the item. If no classification has been made by any of the specialist classifiers, specializing invokes the catch-all classifier, which then assigns a class to the item. This method of combining classifiers helps to ensure that the specializing classifier gives due consideration to all of the classes.

Note that having the specialist classifiers run in a strict order and allowing each specialist to label only those samples that have not been labeled by the earlier specialists is a conflict resolution strategy that allows the specialists for the smaller classes to override the specialists for the larger classes. Similarly, all of the specialist classifiers override the catch-all classifier. We expect that having all classifiers run at the same time, rather than in a strict order, but then imposing a conflict resolution strategy that achieves the same prioritization would give similar results. However, our choice of imposing a strict order on the classifiers eliminates the need for an additional conflict resolution strategy.

3.6. Evaluation metrics and methods

We evaluated specializing using *F*-measures. We compare the specializing classifier to the three complementary J48, Naïve Bayes, and AdaBoost.M1 classifiers and four combined classifiers.

3.6.1. Evaluation metrics

We evaluate the specializing classifier using macro- and micro-averaged *F*-measure [40]. Macro-averaged *F*-measure weights small classes as heavily as larger classes and would more clearly show the change in performance on the small, sparse, less well-represented classes. Micro-averaged *F*-measure gives equal weight to each sample, rather than each class, and gives a sense of overall performance on the complete data.

Macro- and micro-averaged *F*-measures are often used as performance metrics in natural language processing tasks [38,41]. They are computed from precision, recall, and *F*-measure which require counts for true positives (TP), false positives (FP), and false negatives (FN). Precision measures the percentage of the correct assignments made to each class (Eq. (1)). Recall measures the percentage of all items in a class that can be assigned (Eq. (2)). *F*-measure is the harmonic mean of precision and recall (Eq. (3)). Using β, it can favor either precision or recall. In this paper, we give equal weight to precision and recall by setting $\beta = 1$.

$$\text{Precision} = P = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = R = \frac{TP}{TP + FN} \tag{2}$$

$$F\text{-measure} = F = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R} \tag{3}$$

$$\text{Macro-averaged Precision} = P_{macro} = \frac{\sum_{i=1}^{M} P_i}{M} \tag{4a}$$

$$\text{Macro-averaged Recall} = R_{macro} = \frac{\sum_{i=1}^{M} R_i}{M} \tag{4b}$$

$$\text{Macro-averaged F-measure} = F_{macro} = \frac{\sum_{i=1}^{M} F_i}{M} \tag{4c}$$

$$\text{Micro-averaged Precision} = P_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FP_i)} \tag{5a}$$

$$\text{Micro-averaged Recall} = R_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FN_i)} \tag{5b}$$

$$\text{Micro-averaged F-measure} = F_{micro} = \frac{(1 + \beta^2) \times P_{micro} \times R_{micro}}{(\beta^2 \times P_{micro}) + R_{micro}} \tag{5c}$$

Macro-averaged precision, recall, and *F*-measure (Eq. (4)) take the arithmetic mean of the precision, recall, or *F*-measure metrics of all classes. By dividing the sum of a given metric over all classes by the number of classes, M, it gives equal weight to each class, regardless of its size. In contrast, micro-averaged metrics (Eq. (5)) are computed over all samples in the data [38]. When observed together, macro- and micro-averaged metrics give a more complete account of the strengths and weaknesses of systems. Given the focus of this paper on the less well-represented classes, we use macro-averaged F-measure as our primary evaluation metric and study micro-averaged F-measure only to check overall performance on all samples.

3.6.2. Significance testing

We tested the significance of performance differences between classifiers with *Z*-scores. We used a two-tailed test and a *Z*-value of

±1.645. Following the examples of the Message Understanding Conferences [42,43] and of the i2b2 Smoking Challenge Shared-Task [44], we performed all of the significance tests at $\alpha = .10$.

3.6.3. Evaluation methods

We evaluated specializing by comparing the specializing classifier to the three complementary J48, Naïve Bayes, and AdaBoost.M1 classifiers. These three complementary classifiers lie at the heart of our implementation of specializing for multi-class classification of obesity and its co-morbidities. To utilize them as baselines, we trained these classifiers as multi-class classifiers on the training set for each disease, separately on each disease, with the same exact term vector space and attribute vectors that were available to the specializing classifier for each disease. We refer to the resulting classifiers as the *complementary baseline classifiers*. Note that the complementary baselines for a disease are the same set of classifiers from which specializing selects a catch-all classifier for that disease.

We also compared the specializing classifier to four additional *combined baseline classifiers*: voting, stacking, a specialist-only classifier, and a catch-all-only classifier. The combined baseline classifiers were also trained on each disease separately. For each disease, they utilized the training data, term vector space, and attribute vectors available to the specializing classifier for that disease. Voting, as implemented in this paper, combined the three complementary baseline classifiers, J48, Naïve Bayes, and AdaBoost.M1, with majority voting. Stacking combined the three complementary baseline classifiers with the conjunctive rule. The specialist-only classifier for each disease was the combination of the specialist classifiers chosen by the specializing classifier of that disease. The specialist-only classifier ran these specialist classifiers in the same strict order as the specializing classifier. The catch-all-only classifier was selected from the three complementary baseline classifiers for each disease. It was the complementary baseline classifier with the highest micro-averaged *F*-measure on the training set for that disease, i.e., it is the same as the catch-all classifier selected by the specializing method for that disease.

We ran the specializing classifiers and the combined and complementary baselines on each disease separately. We trained each of the specializing and baseline classifiers on the training set for each disease, cross-validated them on the training set of each disease, and finally evaluated them on the test set for each disease. We report results only on the test sets.

We compare the specializing classifier to the combined and complementary baselines in several ways. Although each of the classifiers under study is trained on 16 independent multi-class classification tasks, to understand the overall strengths of these classifiers, we aggregate the performance of each of the classifiers over all classes of all diseases and obtain a global performance measure for each classifier on the task of classifying 16 diseases as present, absent, questionable, or unmentioned (Table 3). In order to understand the strength of each of the classifiers on each of the present, absent, questionable, and unmentioned classes, we measure performance on each of these classes. For this, we aggregate for each of the classes across the 16 diseases, but keep the classes themselves separate (see Table 4). We measure macro-average F-measure on each disease by aggregating the results over the present, absent, questionable, and unmentioned classes of each disease but keeping the diseases themselves separate (Table 5). Finally, we determine the contribution that the specialists and the catch-all classifiers each make to specializing. For each disease, we measure the number of samples classified and the percent correct by the specialists and the catch-all classifiers (Table 6).

Table 3
Aggregate results for classifying obesity and 15 of its co-morbidities by combined and complementary classifiers.

Classifier	Micro-averaged			Macro-averaged		
	Precision	Recall	*F*-measure	Precision	Recall	*F*-measure
Specializing	0.9230	0.9230	0.9230	0.6369	0.4972	0.5229
Specialist-only	0.9276	0.9162	0.9218	0.6348	0.4879	0.5160
Catch-all-only	**0.9159**	**0.9159**	**0.9159**	**0.5330**	**0.4738**	**0.4863**
Voting	0.9175	**0.9147**	0.9161	**0.9505**	**0.4484**	**0.4494**
Stacking	0.9183	0.9183	0.9183	**0.9519**	**0.4495**	**0.4505**
J48	**0.9070**	**0.9070**	**0.9070**	**0.5193**	0.4846	**0.4959**
Naïve Bayes	**0.7291**	**0.7291**	**0.7291**	**0.6618**	**0.3604**	**0.3616**
AdaBoost.M1	**0.9046**	**0.9046**	**0.9046**	**0.9484**	**0.4345**	**0.4402**

Bold indicates statistically significant difference from the specializing classifier. Specializing combines complementary classifiers trained in one-versus-all and multi-class manners in order to create one specializing classifier per disease. Specialist-only classifier for a disease invokes only the specialist classifiers for the present, absent, questionable, and unmentioned classes of the disease; it employs these specialist classifiers in the same strict order used by the specializing classifier. Catch-all-only, voting, and stacking combine the complementary classifiers trained in multi-class manner in various ways in order to create one catch-all-only, one voting, and one stacking classifier per disease. The complementary classifiers J48, Naïve Bayes, and AdaBoost.M1 are also evaluated on their own, after being trained in multi-class classification of each disease separately. Aggregate results report performance on all classes on the complete set of obesity and 15 of its co-morbidities.

4. Results and discussion

The results in Tables 3–5 show that specializing can perform significantly better than the baselines in classifying obesity and 15 of its co-morbidities. Table 6 shows that the specialist classifiers are responsible for majority of the success of specializing. The performance gain of specializing comes, in part, from its ability to perform better on the less well-represented absent class.

4.1. Aggregate result analysis

Overall, specializing improved macro-averaged *F*-measure without adversely affecting micro-averaged *F*-measure. The results in Table 3 show that the specializing classifier demonstrated significant improvement in macro-averaged *F*-measure over all of the complementary baselines and all but one of the combined baselines. The same table shows that the specializing classifier demonstrated significant improvement in micro-averaged *F*-measure over one of the combined baselines and all of the complementary baselines. In both macro- and micro-averaged *F*-measures, the specializing classifier and the specialist-only baseline were not significantly different from each other. In other words, the specialists were responsible for most of the success of specializing and the gain provided by the catch-all classifiers was not statistically significant. On the other hand, the specializing classifier was significantly different from the catch-all-only baseline in both macro- and micro-averaged F-measure. In other words, the catch-all classifiers gained significantly from the addition of specialists.

Comparing the combined baseline classifiers to each other shows that the specialist-only classifier gives the best macro- and micro-averaged results. In other words, specialist-only classifier gets more of the less well-represented classes correct over all diseases (as reflected by macro-averaged metrics) and it also gets the highest raw number of correct assignments (as reflected by micro-averaged metrics). The catch-all-only classifier has the second best macro-averaged results while the second best micro-averaged results come from stacking.

Comparing the specialist-only classifier with the complementary baselines shows that the specialist-only classifier outperforms

Table 4
Performance per class, aggregated over all diseases.

	Predicted				Totals		$Precision_{micro}$	$Recall_{micro}$	$F\text{-}Measure_{micro}$
	Y	N	Q	U					
(a) Specializing									
Y	1891	3	7	296	2197	Y	0.8694	0.8607	0.8651
N	34	11	0	20	65	N	0.7333	0.1692	0.2750
Q	15	0	0	7	22	Q	0.0000	0.0000	0.0000
U	235	1	3	5542	5781	U	0.9449	0.9587	0.9517
Totals	2175	15	10	5865					
(b) Specialist-only									
Y	1848	3	0	291	2197	Y	0.8787	**0.8411**	0.8595
N	26	10	0	17	65	N	0.7143	0.1538	0.2532
Q	15	0	0	7	22	Q	0.0000	0.0000	0.0000
U	214	1	3	5531	5781	U	0.9461	0.9568	0.9514
Totals	2103	14	3	5846					
(c) Catch-all-only									
Y	1865	7	8	317	2197	Y	0.8579	0.8489	0.8534
N	36	6	0	23	65	N	**0.3333**	0.0923	**0.1446**
Q	15	0	0	7	22	Q	0.0000	0.0000	0.0000
U	258	5	2	5516	5781	U	0.9408	0.9542	0.9474
Totals	2174	18	10	5863					
(d) Voting									
Y	1834	0	0	350	2197	Y	0.8663	**0.8348**	0.8503
N	33	0	0	22	65	N	**1.0000**	**0.0000**	**0.0000**
Q	14	0	0	8	22	Q	**1.0000**	0.0000	0.0000
U	236	0	0	5543	5781	U	**0.9358**	0.9588	0.9472
Totals	2117	0	0	5923					
(e) Stacking									
Y	1832	0	0	365	2197	Y	0.8736	**0.8339**	0.8533
N	43	0	0	22	65	N	**1.0000**	**0.0000**	**0.0000**
Q	15	0	0	7	22	Q	**1.0000**	0.0000	0.0000
U	207	0	0	5574	5781	U	**0.9340**	0.9642	0.9488
Totals	2097	0	0	5968					
(f) J48									
Y	1846	17	8	326	2197	Y	**0.8349**	**0.8402**	**0.8376**
N	36	10	0	19	65	N	**0.3030**	0.1538	0.2041
Q	15	0	0	7	22	Q	0.0000	0.0000	0.0000
U	314	6	2	5459	5781	U	0.9394	**0.9443**	**0.9419**
Totals	2211	33	10	5811					
(g) Naïve Bayes									
Y	1342	3	0	852	2197	Y	**0.5142**	**0.6108**	**0.5584**
N	20	3	0	42	65	N	**0.3000**	**0.0462**	**0.0800**
Q	6	0	0	16	22	Q	**1.0000**	0.0000	0.0000
U	1242	4	0	4535	5781	U	**0.8329**	**0.7845**	**0.8079**
Totals	2610	10	0	5445					
(h) AdaBoost.M1									
Y	1686	0	0	511	2197	Y	0.8818	**0.7674**	**0.8206**
N	41	0	0	24	65	N	**1.0000**	**0.0000**	**0.0000**
Q	14	0	0	8	22	Q	**1.0000**	0.0000	0.0000
U	171	0	0	5610	5781	U	**0.9118**	**0.9704**	**0.9402**
Totals	1912	0	0	6153					

Y, present; N, absent; Q, questionable; and U, unmentioned. Bold indicates significant difference from the Specializing Classifier. Note that one specializing, one voting, one stacking, and one of each of multi-class complementary classifiers J48, Naïve Bayes, and AdaBoost.M1 is generated per disease. Each generated classifier can predict any of the classes Y, N, Q, or U. The results in this table are aggregated over all of the classes of all diseases in the data, giving equal weight to each sample regardless of its class, e.g., table (a) for specializing presents the aggregate performance for classes Y, N, Q, and U of specializing classifiers over all diseases.

all of the complementary baselines. In other words, picking and applying the specialist classifiers in a strict order for each disease gives better performance than applying any single complementary baseline to all of the diseases.

Similarly, comparing the catch-all-only baseline with the complementary baselines shows that the catch-all-only classifier outperforms all of the complementary baselines. In other words, picking and applying the best complementary baseline classifier for each disease gives better performance than applying any single complementary baseline to all of the diseases.

Finally, Table 3 also reveals that the complementary baselines differ from each other in their performance. J48 is the strongest and Naïve Bayes is the weakest among them.

Table 4 shows the performance of specializing and the baselines on the present, absent, questionable, and unmentioned classes in the complete test data. On the present and unmentioned classes, which contained the bulk of the judgments, all of the complementary baseline classifiers correctly predicted a majority of the items. The ability to correctly assign test records to the present and unmentioned classes shows that the training set provided sufficient informative samples for all three complementary baseline classifiers to train on for these two classes. As a result, all of the combined baseline classifiers performed well and were able to make correct predictions for these two classes. The specializing classifiers' *F*-measures did not differ significantly from the *F*-measures of the combined baselines on these classes.

Table 5
Macro-averaged *F*-measure by disease.

	Classes with > 0 samples per disease (observed classes)	Specializing	Specialist-only	Catch-all-only	Voting	Stacking	J48	Naïve Bayes	AdaBoost.M1
Asthma	4 (Y, N, Q, U)	0.4586	0.4486	0.4661	0.4637	0.4661	0.4565	**0.3007**	0.4661
Atherosclerotic CV disease	4 (Y, N, Q, U)	0.4168	0.4140	0.3900	0.4055	0.4027	0.3900	0.4135	**0.3619**
Heart failure	4 (Y, N, Q, U)	0.4360	0.4353	0.4280	0.4329	0.4346	0.4092	**0.3774**	0.4280
Depression	2 (Y, U)	0.7150	0.7150	0.7150	0.7035	**0.4618**	0.7387	**0.5161**	0.7150
Diabetes mellitus	4 (Y, N, Q, U)	0.5853	0.5813	0.5708	**0.4461**	**0.4384**	0.5708	**0.3645**	**0.4152**
Gallstones/Cholecystectomy	3 (Y, N, U)	0.6122	0.6192	0.6100	0.5913	0.6054	0.6100	**0.3709**	0.5859
GERD	4 (Y, N, Q, U)	0.4427	0.4427	0.4355	0.4111	0.4435	0.4355	**0.2705**	0.4105
Gout	2 (Y, U)	0.9096	0.9085	0.9096	0.9096	0.9160	0.9160	**0.5267**	0.9096
Hypercholesterolemia	4 (Y, N, Q, U)	0.5393	**0.4885**	**0.4843**	**0.4234**	**0.4094**	**0.4843**	**0.3799**	**0.4094**
Hypertension	4 (Y, N, Q, U)	0.6115	0.6115	**0.4288**	**0.4130**	**0.4288**	**0.5515**	**0.2513**	**0.4288**
Hypertriglyceridemia	2 (Y, U)	0.4940	0.4940	0.4940	**0.5864**	0.4950	**0.5784**	0.4940	**0.6627**
Osteoarthritis	2 (Y, U)	0.8813	0.8813	0.8813	0.8874	0.8813	0.8813	**0.5662**	**0.8145**
Obesity	4 (Y, N, Q, U)	0.4718	0.4718	0.4718	0.4655	0.4718	0.4644	**0.3248**	0.4718
Obstructive sleep apnea	3 (Y, Q, U)	0.6014	0.6014	0.6009	0.5967	0.6170	0.6009	**0.3447**	0.5765
Peripheral vascular disease	2 (Y, U)	0.8452	0.8452	0.8452	0.8264	0.8641	0.8452	**0.6616**	**0.7951**
Venous insufficiency	2 (Y, U)	0.4945	0.4945	0.4945	**0.6607**	0.4950	**0.6514**	0.4945	**0.6607**

Macro-averaged *F*-measure of combined and complementary baseline classifiers for each of obesity and its 15 co-morbidities. One of each kind of classifier is trained per disease. Macro-averages per disease are computed over the classes that are observed in the data for each disease, e.g., Y, N, Q, and U. The classes observed for each disease in the test data are shown in column 2. Bold indicates significant performance difference from the specializing classifier.

Table 6
Assignments by specialists and catch-all classifiers by disease.

Disease	Samples assigned				
	Specialists		Catch-all		Total
	Number assigned	# (%) Correct	Number assigned	# (%) Correct	Number samples
Asthma	496	475 (96)	9	9 (100)	505
Atherosclerotic CV disease	454	385 (85)	44	22 (50)	498
Heart failure	494	428 (87)	5	3 (60)	499
Depression	507	448 (88)	0	N/A	507
Diabetes mellitus	481	442 (92)	23	16 (70)	504
Gallstones/cholecystectomy	504	484 (96)	4	0 (0)	508
GERD	505	480 (95)	0	N/A	505
Gout	503	489 (97)	3	1 (33)	506
Hypercholesterolemia	499	425 (85)	4	4 (100)	503
Hypertension	505	455 (90)	0	N/A	505
Hypertriglyceridemia	508	496 (98)	0	N/A	508
Osteoarthritis	503	467 (93)	0	N/A	503
Obesity	494	464 (94)	0	N/A	494
Obstructive sleep apnea	497	479 (96)	7	0 (0)	504
Peripheral vascular disease	508	475 (94)	0	N/A	508
Venous insufficiency	508	497 (98)	0	N/A	508
Total	7966	7389 (93)	99	55 (56)	8065

Number of samples assigned, and the number and percent assigned correctly, by the specialists and catch-all classifiers for each of obesity and its 15 co-morbidities.

The less well-represented questionable class showed the opposite of the observations from the present and unmentioned classes. An examination of the results revealed that neither specializing nor any of the baselines was able to correctly classify discharge summaries judged as questionable. The questionable class was the least well-represented class, with just 39 out of 11,630 instances in the training data (see Table 1). None of the complementary baseline classifiers were able to correctly classify any of the 22 discharge summaries marked as questionable in the test data (see Table 4). Since each of the combined baseline classifiers relied upon the accuracy of the complementary classifiers, none of the combined baselines were able to correctly classify the questionable class. We attribute the results on the questionable class to the lack of a sufficient number of informative samples for this class in the training set. In Table 1, for example, we observe that for two of the diseases, Heart failure and Hypertension, there were no training samples for the questionable class even though those diseases were associated with five of the 22 discharge summaries judged as questionable in the test set.

The second least well-represented class, the absent class, revealed the difference in the strengths of the various classifiers. The absent class consisted of just 87 of the 11,630 judgments in the training data (see Table 1). Both the J48 and Naïve Bayes complementary baseline classifiers made some correct predictions (10 and 3 true positives, respectively) for this class. This, in turn, enabled the specializing, specialist-only, and catch-all-only classifiers to correctly predict the judgments, at least some of the time, for this class.

On the absent class, the specializing classifier showed significant improvement in *F*-measure over three of the combined baselines (see Table 4). Its difference from the specialist-only baseline was not statistically significant, i.e., most of specializing's gain on this class comes from its specialists. The catch-all-only baseline was successful in recognizing the absent class part of the time. The fact that the catch-all-only classifier is able to get some of these samples correct implies that the complementary baseline classifiers that constitute the catch-all classifiers can capture this information.

On the other hand, voting and stacking cannot directly benefit from the catch-all classifiers (i.e., the best complementary classifier per disease) in predicting the absent class. This is because the combination of the catch-all classifier for a disease with the rest of the complementary baseline classifiers for that disease, through either voting or stacking, overrides the informative catch-all classifier for the disease with the incorrect predictions provided by the remaining complementary baseline classifiers for that disease. The catch-all-only classifiers apply the catch-all classifier for each disease directly (rather than combining it with the rest of the complementary baseline classifiers) and accept the catch-all classifier as the authority on the items that the catch-all classifier gets to label. This contributes to the catch-all-only classifiers' gain in performance over voting and stacking on the absent class.

4.2. Disease-level results analysis

Given the performance of the classifiers on the overall task of classifying diseases, we compared the classifiers at the disease level for each of the 16 diseases (see Table 5). For this, we obtained aggregate results over the present, absent, questionable, and unmentioned classes of each disease. Note that the number of classes containing samples, i.e., the number of classes that were actually observed for each disease, varied from disease to disease in our data. For those diseases that had samples in more than two classes, i.e., were in fact multi-class classification problems, we found the specializing classifier performed no worse than, and at times showed significant improvement over, all of the baselines.

For three of the diseases, Diabetes mellitus, Hypercholesterolemia, and Hypertension, the specializing classifier outperformed at least four of the seven baseline classifiers. These three diseases can be distinguished from the remaining diseases by virtue of the number of classes observed in their training data and by the rate of true positives predicted by the complementary classifiers. Specifically, these three diseases all have multi-class judgments and at least one of the complementary baseline classifiers' recall was above .10 for all of the classes, excluding the questionable class. The remaining thirteen diseases fail to meet these criteria.

In other words, the specializing classifier showed significant performance improvement over, or performed no worse than, the combined baselines on the diseases that contained at least three classes to predict, as long as those classes could reliably be predicted by at least one of the complementary classifiers.

Most of the success of specializing at the disease level was accounted for by the specialists. As detailed by disease in Table 6, for eight of the 16 diseases, all of the samples in the test set were classified by the specialist classifiers, i.e., the catch-all classifier was not invoked. For six of the remaining eight diseases, less than 2% of the samples invoked the catch-all classifier. An examination of those records not classified by the specialists indicates that they may be difficult to classify. For example, for record 1020 the ground truth listed both Asthma and Diabetes as being present in the patient, but the specialists for both Asthma and Diabetes did not classify this record. While the text "asthma" was present in record 1020, other indicators of asthma as identified by the specialist classifier, such as "inh" (as in "2 puff inh"), were missing from this record. Similarly, while the text "diabetes" appeared in record 1020, other indicators of diabetes as identified by the specialist classifier, such as "insulin", were missing from this record.

4.3. Specializing analysis and discussion

All of our results indicate that the specialists are responsible for most of the success of the specializing classifiers. Our analysis of the data samples in the test set showed that the contribution of the catch-all classifiers to specializing was limited by the number of samples passed on to the catch-alls by specializing. As a component of specializing, the catch-all classifiers predicted only those samples that failed to receive a definitive class assignment from any of the specialists for a disease, i.e., those samples that could not be classified by any of the specialist classifiers for a disease. Table 6 shows that only 99 of the 8065 test samples over all diseases were passed on to the catch-all classifiers by specializing.

For five of the 16 diseases, the training data contained only two classes, present and unmentioned, with which to train specialist classifiers (see Table 1). The specialists trained under these conditions were complementary and exhaustive in the classification of their disease, i.e., for any given sample, exactly one of the two specialists predicted a class. This left no samples for the catch-all classifiers for these diseases.

While the contribution of the catch-all classifiers in specializing is limited, the catch-all classifiers themselves are not weak (see Table 3); they can even recognize samples from the less well-represented absent class (see Table 4). What is more, the catch-all classifiers can complement the specialist classifiers on some data samples, e.g., in record 1020, they can correctly classify samples passed over by the specialists with respect to asthma and diabetes. A difference in the relative employment order of the specialists and the catch-all could change the number of samples classified by the catch-all.

Extrapolating this hypothesis to the specialists, we checked if a change in the order of the specialists could affect overall performance. For each of the diseases, we examined the specialists in a pair-wise manner (e.g., present-absent, absent-unmentioned). We compared the predictions that would have been made by each of the specialists in the pair, had they each been asked to predict the sample. In the vast majority of the cases, no more than one of the specialists made a definitive prediction for a given sample. Across 16 diseases, we found only 80 instances where both of the specialists would have made definitive, and therefore conflicting, predictions. In order to see the impact of those 80 instances upon classification, we examined all of the combinations for ordering the specialists. We found that, for all 16 diseases, the order specified by specializing was either significantly better than or not significantly different from any other order of the specialists. Each case where a significant difference was found involved the placement of the specialist for the less well-represented absent class. For example, for the disease Hypertension, any combination that invoked the absent class before the present class performed significantly better than those combinations that invoked the present class before the absent class.

5. Limitations and future work

Specializing treats the classification of each disease as an independent multi-class classification task. For each disease, only one class is assigned to a record, i.e., the choice of one class excludes the remaining classes. Additional study will be necessary to apply specializing to tasks that require assignment of multiple labels to each data item.

Specializing utilizes the best OVA-trained classifier for each class given the decisions already made by the specialists for the less well-represented classes. While we have found that specializing's combination of specialists provides the best performance from among all combinations of specialists, there may exist other combinations of OVA-trained classifiers that give suboptimal performance on some classes for the sake of achieving globally optimal results on the complete data.

While our results are encouraging, future work should look into ways of improving the performances of the catch-all classifiers and of specializing. We hypothesize that the performance of catch-all classifiers can be improved by training the catch-all classifiers on

the samples that could not have been correctly classified by the specialists (currently, specializing trains the catch-all classifiers on the complete training data). We expect that the samples that fail to be classified by the specialists are harder to classify and training catch-all classifiers on these samples could improve their contribution to the specialists. Alternatively, the catch-all classifiers could be replaced by more OVA-trained classifiers, e.g., the OVA-trained classifiers that performed second best and were not selected as specialists but demonstrated potential. To this end, including a threshold for minimum required performance level from the specialists and from any of the OVA-trained classifiers may further contribute to performance. These hypotheses remain untested mostly due to limitations of the data set. A review of the quantity and distribution of the samples that were left for the catch-all classifiers indicated the lack of a sufficient number of samples that could satisfactorily and reliably be used as training data for the experiments required to test these hypotheses.

Finally, future work should study the generalizability of specializing, by applying specializing to corpora other than medical discharge summaries and to corpora with more than four classes, possibly with alternative sets of complementary classifiers.

6. Accessibility of data

We have presented and studied specializing on a data set generated for the i2b2 Shared-Task and Workshop on Challenges in Natural Language Processing for Clinical Data: Obesity Challenge [3]. This data has been made available to the research community from i2b2.org/NLP under a data use agreement, for studies that relate to the i2b2 Shared-Task.

With few exceptions, such as the challenge data released by the University of Cincinnati Computational Medicine Center [45], the medical language processing community is haunted by the lack of a set of publicly available records that have been annotated for the gold standard, that can serve as a test bed for the development of competitive or complementary systems, and that can help replicate past work and advance the state of the art. This limitation stems from legal, proprietary, and privacy concerns [46] that make the patient records available only to limited audiences and under strict confidentiality agreements.

Although currently it is not a substitute for a public domain data set that would offer greater benefits to the research community, the data set used in this study takes a step towards remedying this problem. While we appreciate the need for a public domain data set and aspire to help create it, until we obtain the institutional approvals that would make this aspiration a reality, we invite the research community to participate in future shared-tasks that would grant them access to similar data. We also wholeheartedly support complementary activities that would bring about public domain medical record data sets for use by the research community.

7. Conclusions

We examine the performance of specializing based on its ability to predict whether obesity and each of 15 of its co-morbidities are present, absent, or questionable in a patient, or unmentioned in the patient discharge summary. For each disease, specializing combines multi-class classifiers trained in an OVA manner with a catch-all classifier that is trained as a multi-class classifier that can distinguish all classes from one another. The specializing approach to combining classifiers is an effective way to improve macro-averaged performance in multi-class classification of diseases in the presence of less well-represented classes with a sufficient minimum number of samples to learn from based on medical discharge summaries. This improvement is accomplished without sacrificing overall micro-averaged performance.

Acknowledgments

This work was supported in part by the National Institutes of Health through research grants 1 RO1 EB001659 from the National Institute of Biomedical Imaging and Bioengineering and through the NIH Roadmap for Medical Research, Grant U54LM008748.

We gratefully acknowledge Chen Song for his tireless programming assistance, and Peter Szolovits and the two anonymous reviewers of the Journal of Biomedical Informatics for their constructive and insightful comments.

References

[1] Sibanda TC, He T, Szolovits P, Uzuner Ö. Syntactically-informed semantic category recognition in discharge summaries. In: Proceedings of the AMIA symposium; 2006. p. 714–8.
[2] Goldstein I, Arzumtsyan A, Uzuner Ö. Three approaches to automatic assignment of ICD-9-CM codes to radiology reports. In: Proceedings of the AMIA symposium; 2007. p. 279–83.
[3] Second i2b2 shared-task and workshop: challenges in natural language processing for clinical data. Available from: http://www.i2b2.org/NLP/.
[4] Krenzelok E, Macpherson E, Mrvos R. Disease surveillance and nonprescription medication sales can predict increases in poison exposure. J Med Toxicol 2008;4:7–10.
[5] Jain AK, Chandrasekaran B. Dimensionality and sample size considerations in pattern recognition practice. In: Krishnaia PR, Kanal LN, editors. Handbook of statistics. Amsterdam, The Netherlands; 1982, p. 835–55.
[6] Tang L, Liu H. Bias analysis in text classification for highly skewed data. In: Fifth IEEE international conference on data mining (ICDM'05). Houston, TX; 2005. p. 781–4.
[7] Wolpert DH. Stacked generalization. Neural Netw 1992;5:241–59.
[8] Freund Y. Boosting a weak learning algorithm by majority. Inf Comput 1995;121:256–85.
[9] Schapire RE. The strength of weak learnability. Mach Learn 1990;5:197–227.
[10] Schapire RE, Singer Y. BoosTexter: a boosting-based system for text categorization. Mach Learn 2000;39:135–68.
[11] Chan PK, Stolfo SJ. Experiments on multistrategy learning by meta-learning. In: Proceedings of the second international conference on Information and knowledge management. Washington, DC: ACM; 1993.
[12] Chan PK, Stolfo SJ. Toward parallel and distributed learning by meta-learning. In: AAAI workshop in knowledge discovery in databases; 1993. p. 227–40.
[13] Chan PK, Stolfo SJ. A comparative evaluation of voting and meta-learning on partitioned data. In: Prieditis A, Russell S, editors. Proceedings of the twelfth international conference on machine learning. San Francisco: Morgan Kaufman; 1995.
[14] Zenko B, Todorovski L, Dzeroski S, Cercone N, Lin TY, Wu X, Cercone N, Lin TY, Wu X. A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods. In: Proceedings 2001 IEEE international conference on data mining. San Jose, CA: IEEE Comput. Soc.; 2001.
[15] Liu Y-Z, Jiang Y-C, Liu X, Yang S-L. CSMC: a combination strategy for multi-class classification based on multiple association rules. Knowledge-based systems 2008;21:786–93.
[16] Daskalakis A, Kostopoulos S, Spyridonos P, Glotsos D, Ravazoula P, Kardari M, et al. Design of a multi-classifier system for discriminating benign from malignant thyroid nodules using routinely H&E-stained cytological images. Comput Biol Med 2008;38:196–203.
[17] Eom J-H, Kim S-C, Zhang B-T. AptaCDSS-E: a classifier ensemble-based clinical decision support system for cardiovascular disease level prediction. Expert Syst Appl 2008;34:2465–79.
[18] Duin RPW, Tax DMJ, Kittler J, Roli F, Kittler J, Roli F. Experiments with classifier combining rules. In: Multiple classifier systems. Proceedings of first international workshop, MCS 2000. Lecture Notes in Computer Science, vol.1857. Berlin, Germany: Springer-Verlag; 2000.
[19] Tax DMJ, van Breukelen M, Duin RPW, Josef K. Combining multiple classifiers by averaging or by multiplying? Pattern Recognit 2000;33:1475–85.
[20] Hu R, Damper RI. A 'No Panacea Theorem' for classifier combination. Pattern Recognit 2008;41:2665–73.
[21] Kuncheva LI. Combining pattern classifiers: methods and algorithms. Hoboken, NJ: J. Wiley; 2004.
[22] Porter M. An algorithm for suffix stripping. Program 1980;14:130–7.
[23] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manage 1988;24:513–23.
[24] Witten IH, Frank E. Data mining: practical machine learning tools and techniques. 2nd ed. Amsterdam, Boston, MA: Morgan Kaufman; 2005.
[25] Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of the AMIA symposium; 2001. p. 17–21.

[26] Rao RB, Sandilya S, Niculescu RS, Germond C, Rao H. Clinical and financial outcomes analysis with existing hospital patient records. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining: ACM Press: New York, NY, USA; 2003. p. 416–25.
[27] Kim JJ, Park JC. Extracting contrastive information from negation patterns in biomedical literature. ACM Trans Asian Language Inf Process 2006;5:44–60.
[28] Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J Am Med Inf Assoc 2001;8:598–609.
[29] Sibanda TC. Was the patient cured? Understanding semantic categories and their relationships in patient records. In: Electrical engineering and computer science. Master's Thesis, Cambridge, MA: Massachusetts Institute of Technology; 2006.
[30] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries; J Biomed Inform 2001. p. 301–10.
[31] Ryan R, Aldebaro K. In defense of one-vs-all classification. J Mach Learn Res 2004;5:101–41.
[32] Bickel PJ, Levina E. Some theory for Fisher's Linear Discriminant function, "naive Bayes", and some alternatives when there are many more variables than observations. Bernoulli 2004;10:989–1010.
[33] Polat K, Güneş S. A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. Expert Syst Appl 2009;36:1587–92.
[34] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55:119–39.
[35] Quinlan JR. Induction of decision trees. Mach Learn 1986;1:81–106.
[36] Quinlan JR. C4.5: programs for machine learning. San Mateo, California: Morgan Kaufman; 1993.
[37] Shannon CE, Weaver W. The mathematical theory of communication. Urbana: University of Illinois Press; 1949.
[38] Yang Y, Liu X. A re-examination of text categorization methods. In: Proceedings of SIGIR international conference on R&D in information retrieval; 1999. p. 42–9.
[39] Freund Y, Schapire RE. Experiments with a new boosting algorithm. In: Machine learning: proceedings of the 13th international conference; 1996. p. 148–56.
[40] Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv 2002;34:1–47.
[41] Salton G, McGill MJ. Introduction to modern information retrieval. New York: McGraw-Hill; 1983.
[42] Grishman R, Sundheim B. Message understanding conference-6: a brief history. In: 16th conference on computational linguistics. Copenhagen, Denmark: Association for Computational Linguistics; 1996. p. 466–71.
[43] Hirschman L. The evolution of evaluation: lessons from the message understanding conferences. Comput Speech Lang 1998;12:281–305.
[44] Uzuner O, Goldstein I, Luo Y, Kohane I. Identifying patient smoking status from medical discharge records. J Am Med Inf Assoc 2008;15:14–24.
[45] Pestian JP, Brew C, Matykiewicz P, Hovermale DJ, Johnson N, Cohen KB, et al. A shared task involving multi-label classification of clinical free text. In: ACL: BioNLP. Prague: Association for Computational Linguistics; 2007. p. 97–104.
[46] Standards for privacy of individually identifiable health information. Department of Health & Human Services; 2000 [45 CFR §160 and 164].