HHS Public Access
Author manuscript
J Biomed Inform. Author manuscript; available in PMC 2016 December 01.

Published in final edited form as:
J Biomed Inform. 2015 December ; 58: 60–69. doi:10.1016/j.jbi.2015.08.019.

Comparison of Machine Learning Classifiers for Influenza Detection from Emergency Department Free-text Reports

Arturo López Pineda[a], **Ye Ye**[a], **Shyam Visweswaran**[a], **Gregory F. Cooper**[a], **Michael M. Wagner**[a], and **Fuchiang (Rich) Tsui**[a,*]

[a]Department of Biomedical Informatics, University of Pittsburgh School of Medicine, 5607 Baum Boulevard, Pittsburgh, PA

Abstract

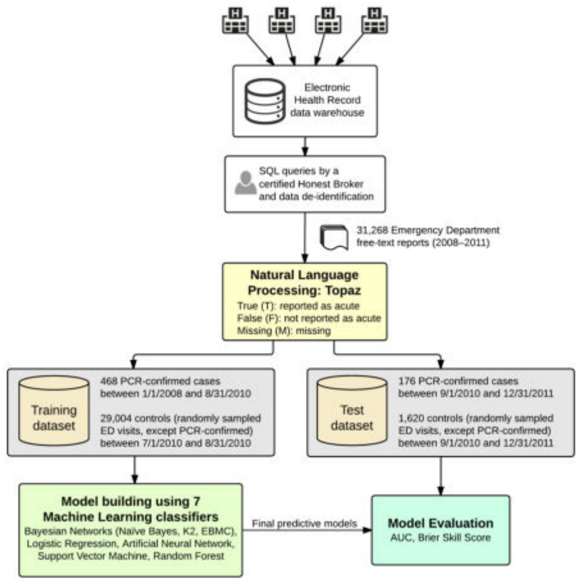
Influenza is a yearly recurrent disease that has the potential to become a pandemic. An effective biosurveillance system is required for early detection of the disease. In our previous studies, we have shown that electronic Emergency Department (ED) free-text reports can be of value to improve influenza detection in real time. This paper studies seven machine learning (ML) classifiers for influenza detection, compares their diagnostic capabilities against an expert-built influenza Bayesian classifier, and evaluates different ways of handling missing clinical information from the free-text reports. We randomly identified 31,268 ED reports from 4 hospitals between 2008 and 2011 to form two different datasets: training (468 cases, 29,004 controls), and test (176 cases and 1,620 controls). We employed Topaz, a natural language processing (NLP) tool, to extract influenza-related findings and to encode them into one of three values: Acute, Non-acute, and Missing. Results show that all ML classifiers had areas under ROCs (AUC) ranging from 0.88 to 0.93, and performed significantly better than the expert-built Bayesian model. Missing clinical information marked as a value of *missing* (not missing at random) had a consistently improved performance among 3 (out of 4) ML classifiers when it was compared with the configuration of not assigning a value of *missing* (missing completely at random). The case/control ratios did not affect the classification performance given the large number of training cases. Our study demonstrates ED reports in conjunction with the use of ML and NLP with the handling of missing value information have a great potential for the detection of infectious diseases.

Graphical abstract

[*]Corresponding author: Fuchiang (Rich) Tsui, Ph.D., *Telephone*: +1 (412) 648-7182, tsui2@pitt.edu.

Ethics approval

The University of Pittsburgh IRB provided ethics approval.



Keywords

Influenza; Emergency Department Reports; Case Detection; Machine Learning; Bayesian

1 Introduction

A key goal for public health epidemiologists is to detect influenza outbreaks early and accurately to save lives and reduce healthcare costs. One way to detect influenza outbreaks earlier is to deploy a public health surveillance (or biosurveillance) system that monitors routinely collected patient data. For example, BioSense [1] is a system developed by the United States Centers for Disease Control and Prevention (CDC), which collects public health information from electronic health records to facilitate regional and national biosurveillance.

Improving the accuracy of disease detection in individual patients is an important element in improving the performance of a biosurveillance system [2]. It is desirable to reduce the time lag to outbreak detection and concurrently retain high accuracy of individual disease detection. To improve timeliness, several modern biosurveillance systems use chief complaints (CCs) from emergency departments (EDs). In contrast to CCs, biosurveillance systems that use laboratory-confirmed reports of diseases have higher diagnostic accuracy and lower false alarm rates but take a longer time to detect outbreaks. For example, it takes on average 1.1 days of turnaround time for the results of polymerase chain reaction (PCR)-based laboratory test for influenza to be available after a visit [3]. Most importantly, many patients with influenza may not have a laboratory test ordered due to costs or hospital policies.

1.1 Data Sources for Biosurveillance

Several approaches have been described for detecting influenza outbreaks. The data sources range from chief complaint with low diagnostic accuracies, to PCR-based laboratory tests [4] with very high diagnostic accuracies.

Hospital laboratory reports of confirmed influenza cases—Using a laboratory test such as PCR for identification of influenza cases has proven to be highly sensitive and specific [5], [6]. Laboratory methods could take quite some time to get results back due to various delays. For example, a culture-based laboratory test takes on average seven days to obtain a result, whereas using real-time PCR (qPCR) reduces the time to about 3–8 hours. We also have to consider the time requirements for specimen preparation, which can vary from minutes to over an hour, depending on whether a manual or automated method is used [7]. If we consider delays in the workflow, overload in the laboratory, transportation of samples, and other human-related factors, it would add extra days to getting a result back. Furthermore, laboratory testing might not even be available to every patient visit due to the incurred laboratory testing cost. For example, in a retrospective evaluation of the 2009 influenza outbreak in Mexico, only 27% of 63,479 patients with influenza-like-illness had access to qPCR testing [8]. As only a small percentage of patients are tested for influenza due to these additional costs, it is not practical to use lab test results for detection of influenza for every patient visit.

Sentinel clinician reports of influenza-like-illness (ILI) cases—The advantage of this approach is that surveillance is performed routinely in all outpatient visits for ILI cases in the sentinel clinics. This approach requires the collaboration of clinicians who might be overburdened by their normal workload. In addition, these reports are not specific to influenza (since other respiratory viruses such as parainfluenza, adenovirus, and respiratory syncytial virus, also cause ILI symptoms). In terms of reporting frequency, this approach is not ideal since the reports are made on a weekly basis. Also, in many cases the reports are generated manually, which may introduce additional errors and time delays in the process [9].

Respiratory or constitutional syndrome monitoring through ED chief complaints (CCs)—The advantages of this approach are the routine collection of CCs in every ED visit and their almost real-time availability. However, using CCs may not be specific enough since many diseases have common symptoms and findings [10], and hence CCs may have low diagnostic accuracy. Increased attention has been given to the use of ED reports for biosurveillance due to the wealth of patient information it provides, including clinician’s diagnoses and treatment [11].

1.2 Machine Learning Classifiers for Influenza Detection

Recently, increased attention has been given to the use of Machine Learning (ML) classifiers for the detection of influenza cases from ED reports. Elkin et al. [12] demonstrated that applying a logistic regression classifier to ED reports has significantly better prediction performance (AUC: 0.764) than applying a model to CCs (AUC: 0.653).

Similarly, Tsui et al. [13] used an influenza-specific expert-constructed Bayesian Network to diagnose influenza in individual patient ED reports achieving an AUC of 0.956.

Both Elkin and Tsui followed a pipeline that first extracts clinical features and maps them to codes using a natural language processing (NLP) tool, then they use a machine learning (ML) classifier to estimate the presence of influenza. Evaluation of this pipeline showed significant differences in performance depending on the NLP tool used [14]. However little attention has been given to the evaluation of the ML classifiers used.

In a preliminary study [15], we compared seven ML classifiers with an expert constructed Bayesian network for detection of patients with influenza syndrome. We used 41,035 ED reports from 8 hospitals and obtained a tie between all ML classifiers with AUCs ranging between 0.97 and 0.99. Nevertheless, the selection of controls for test data used in both studies [13,15] was biased due to the consideration of only ED visits during the summer month of July 2011, which may not represent the true overall case detection performance over a longer time period. However, in this study we randomly selected controls for test data during a period of 18 months: from September 2010 to December 31, 2011. In addition, this study defines a symptom value to be true (T) if the symptom is acute whereas the previous studies only considered if a symptom is “present” for true value; similarly in this study, if a symptom is not acute or absent, the value of the symptom is false (F).

1.3 Handling of Missing Data

One important contribution of this paper is the recommendation on how we should handle missing (i.e., not mentioned) values from ED reports when building ML classifiers for detection of influenza cases. It is common that free- text clinical reports have missing values. Lin and Haug [16] compared the detection performance of only two Bayesian networks using the three missing data categories described below.

Not Missing At Random (NMAR)—The assumption is that missing values in this category cannot be derived from the observed data. If we would like to consider the data (findings or symptoms) that are missing for a specific reason, we could either assign “false” or “missing” for the data. For example, when the value of “nasal swab order” is missing, we may assign “false” if we assume that physician did not write the information about “nasal swab order” when there is no nasal swab order. However, a missing may not always be “false” and it could have many uncertainties: (1) the physician did exam/ask whether the patient has the finding or not; (2) the physician forgot to write it down; (3) a NLP tool failed to extract the information from the report. A conservative choice is to assign missingness to be “missing” instead of “false” which assumes a finding did not occur or not acute..

Missing at Random (MAR)—The assumption is that the absence of a data element depends only on the observed data. This assumption implies that the missing elements have no assigned values, and the missing data can be inferred from observed values.

Missing Completely At Radom, (MCAR)—The assumption is that the absence of data elements is not associated with any other value in the dataset, implying that observing a third state or assigning state False would not introduce additional information. Therefore, there is no need to assign a value for the missing data under this assumption.

1.4 Significance and Contribution

There are still open questions of interest to epidemiologists in health departments. Given a NLP of interest, which machine learning classifier is preferred for detecting influenza cases? Will disease models automatically built by machine learning classifiers perform similar to or better than influenza-specific expert-built models? What is the most appropriate assumption that can be made when dealing with missing data in electronic health records (EHRs)? Choosing between expert-constructed models or automatically learned models is still an open question and in this paper we compare their diagnostic capabilities. We hypothesize that the performance of these automatically constructed models is similar to models built manually by experts.

To the best of our knowledge, our paper is the first one to systematically evaluate the performance of classifiers based on three different missing data category configurations applied to both training and test datasets with Brier score (defined in the Methods section) as an additional metrics, test data resampling, and the use of symptom acuteness for predictive performance evaluation.

2 Materials and Methods

This section describes the NLP, datasets, machine learning classifiers, experimental design, and evaluation metrics used for comparing the performance of the classifiers.

2.1 Topaz

Topaz is a pipeline-based NLP system developed expressly to extract clinical concepts from clinician reports [17], [18]. Previously, Topaz produced a clinical concept with the value of present or absent. In this study, we updated Topaz to output one of three values for a clinical concept: acute, non-acute, or missing, to focus on infectious diseases surveillance.

Topaz processes a report as follows: First, the IndexFinder algorithm [19] maps textual elements to Unified Medical Language System (UMLS) concept unique identifier (CUI) codes [20]. Second, Topaz applies knowledge from the Extended Syndromic Surveillance Ontology (ESSO) [21], [22] to find attribute–value pairings such as “temp 38.5C”, and section-header–term pairs such as “NECK: no tenderness or lymphadenopathy” (which provides evidence that the targeted finding cervical lymphadenopathy is absent). Third, Topaz uses the ConText algorithm [23] to determine who experienced the finding (e.g., patient), whether the finding is mentioned in a conditional way (e.g., “if patient experiences fever, she should return”), and whether the finding is historical (chronic). Lastly, Topaz integrates information from all the values that a finding appears in the report to determine its final output value of acute (state **T**rue), non-acute (state **F**alse), or not mentioned (state **M**issing).

The guidelines used by Topaz for the determination of condition (finding) acuteness are the following. For the non-acute, the condition began more than two weeks ago; meanwhile for the acute: the condition began less than two weeks ago – either a new illness or an acute exacerbation of a chronic illness. This guideline also include the following rules:

1. Discharge diagnoses should be marked as acute as these diagnoses refer to problems that caused a patient to come to the ED.
2. Physical findings, laboratory findings, and radiology findings that are measured or observed during the current clinical encounter should be marked as acute. If they were measured or observed at another visit, annotate them as acute or non-acute depending on when the visit was.
3. If there is not explicit or implicit information in the text about when a clinical condition began, assume it began within the last two weeks and assign the value acute. (This is based on the idea that if they came to the ED with the finding it is most likely a recent event.)
4. Risk factors are often not associated with temporal information. Annotate risk factors as non- acute, because they are assumed to have started more than two weeks ago. For this project, risk factors include the following conditions: smoking, drinking, illicit drug use, patterns in which an organ or location precedes the phrase "risk factors", such as "cardiac risk factors", "stroke risk factors"

2.2 Datasets

Unlike conventional syndromic surveillance, which primarily uses ED CCs that are recorded by triage nurses [10], [24], in this study we used ED reports that have been recorded by clinicians. We collected a total of 31,268 ED reports from four EDs in the University of Pittsburgh Medical Center (UPMC) Health System. The ED reports represent ED visits during the period between 01/01/2008 and 12/31/2011.

To adhere to HIPPA policy in this study, we used a third-party service that follows a standard operation procedure for patient information retrieval from hospital EHR systems and de-identification, known as trusted data brokerage service. The study was governed by an approved IRB protocol (PRO08030129) at the University of Pittsburgh. The process of constructing study datasets for machine learning classifiers was as follows. To de-identify the raw free-text ED reports, a trusted data broker first used the De-ID software [25], which has been approved for such use by the University of Pittsburgh Institutional Review Board (IRB). Then the trusted broker applied Topaz, which extracted clinical findings related to influenza from the de-identified free-text reports. Ye et al. [14] found that Topaz had an overall accuracy of 0.78 to extract findings from free-text ED reports, which was significantly better than the alternative NLP software MEDLEE [26] with an accuracy of 0.71. In Ye's study, the Topaz assigned a finding's value without considering acuteness. In this study, we used Topaz to encode 31 findings with UMLS CUI codes Topaz encoded each finding as either "reported as acute" (value True or T) if it was present in the report as being acute, or not "reported as acute" (value False or F) if either the finding was specifically reported as not being present or it was reported as being chronic. The rest of the findings were encoded as "missing" (value Missing or M) if they were not found in the report.

The expert-constructed influenza model comprises 31 influenza-related findings based on the experience of the board-certified domain expert's professional assessment [14]. Topaz was configured to extract these 31 clinical findings from ED reports. From the 31 findings targeted by Topaz some might be missing in a report. For example, a clinician may not mention that a patient has cough or rule out cough and not write it down in the report; thus Topaz reports the finding cough as missing in the output.

Table 1 lists 31 features used by the expert model and our machine learning classifiers; they are ranked in the descending order by likelihood ratio. Eq. (1) defines the likelihood ratio, where $P(D+)$ represents the prevalence of influenza, $P(T+|D+)$ is the conditional probability of the feature being True given that influenza is True, and P(T+|D−) is the conditional probability of the feature being True given that influenza is False.

$$likelihood\ Ratio=\frac{P(T+|D+)}{P(T+|D-)} \quad (1)$$

We constructed a training dataset and a test dataset for the study from the UMLS CUI codes produced by Topaz. The training dataset set consisted of 468 PCR-confirmed cases of influenza between 1/1/2008 and 8/31/2010, and 29,004 controls (all ED visits in the summer between 7/1/2010 and 8/31/2010, excluding PCR-test positives). The test dataset consisted of 176 PCR-confirmed cases and 1,620 randomly sampled controls from ED visits between 9/1/2010 and 12/31/2011. The sampling time period includes influenza seasons in the years 2010 and 2011. Estimating the real prevalence of an influenza outbreak prospectively is a task with a high degree of uncertainty and requires a better understanding of population disease models [27]. The training dataset had a low prevalence of influenza (1.6%) that could resemble the prevalence rate of a non-influenza period; while we used for testing a higher prevalence of influenza (9.8%), which emulates the conditions of a hypothetical outbreak and it is also close to the prior setup in the expert BN model. Figure 1 shows the study process flow.

2.3 Expert-constructed Classifier

In our previous study by Tsui et al. [13], a board-certified infectious disease domain expert and his two colleagues assessed a Bayesian network that consists of a set of nodes representing influenza and its findings. An arc between two nodes represents a probabilistic dependency. The prior probability for influenza was set to 10%, and the network is composed of 31 nodes that have a nearly naïve Bayes structure. The nodes include features such as fever, cough, nausea, headache, wheezing, chill, etc. (Table 1 shows a complete list). Compared with Elkin's influenza predictive model [12], this classifier used 21 more findings for influenza modeling. The expert-constructed classifier was built using a software tool, the Graphical Network Interface (GeNIe) [28]. Figure 2 illustrates a graphical representation of the model.

2.4 Machine Learning Classifiers

The ED reports provide a wealth of patient visit information, and were used to build machine learning classifiers to detect influenza cases. To build the classifiers we used

WEKA's (Waikato Environment for Knowledge Acquisition, version 3.6 [29]) and compare them to an expert-built Bayesian network (BN) [30] influenza model.

The following summarizes each of the classifiers used in this study.

Expert-MLE—We used the structure of the expert-built BN and modified the network probabilities by using maximum likelihood estimates derived from a training dataset.

Naïve Bayes (NB) [31]—NB is a simple BN classifier that assumes that feature (finding) nodes are conditionally independent of each other given the target node. The probability parameters are estimated from the training data. The parameters were estimated from data using the maximum likelihood estimator.

Bayesian Network with the K2 algorithm (K2-BN) [32]—This classifier learns a BN from the data using the K2 scoring function and search method to evaluate the probabilities of a node having a specific parent or set of parents. It is a forward hill-climbing classifier that iteratively evaluates potential models. We set the maximum number of parents for a node to 31, allowing the algorithm to search over all possible parent configurations for a given node. This value is reasonable given the small number of features. The conditional probabilities were estimated using a maximum likelihood estimator. The order of the nodes was set in decreasing order of likelihood ratios followed by the disease (influenza) node.

Efficient Bayesian Multivariate Classification (EBMC) [33], [34]—This classifier performs greedy search in a subspace of BNs to find the one that best predicts a target node. It initially starts with an empty model and then it identifies a set of nodes that are parents of the target and predicts it well. EBMC then transforms the temporary network structure into a statistically equivalent one where the parents of the target become children of the target with arcs among them. Next, it greedily eliminates arcs among these children that improve the prediction of the target. It then iterates the whole process until no set of parents (which we can view as a "probabilistic rule") can be added to the target node to improve the prediction of it. The expected number of predictors was set to 31, and the models were evaluated using the K2 scoring measure. EBMC was implemented as an independent classifier in the Java programming language.

Logistic Regression (LR) [35]—It is a parametric classifier that learns a function of the form P (Y|X), where Y is the target class (such as a disease), and X is a vector of input values. To improve feature estimation, a penalized log likelihood function is used. We built a multinomial logistic regression model with a ridge penalty in the likelihood function of 1.0E-8 (default) and iteration was done until convergence.

Artificial Neural Networks (ANN) [36]—It is a flexible model that expresses complex non-linear relationships among the features, which consists of an interconnected group of variables. In a basic ANN model there are three layers of neurons that can learn from data iteratively through a back propagation classifier. The backpropagation classifier was used to train a multilayer perceptron with one hidden layer, and with the number of nodes equal to the sum of features and classes. We assigned Weka's default values for a learning rate with decay of 0.3, and a momentum rate for the backpropagation classifier of 0.2. Suitable ranges for these parameters have been found to be between 0.15 – 0.8 for learning rate, and 0.1 – 0.4 for momentum [37].

Support Vector Machine (SVM) [38]—A basic SVM is a non-probabilistic linear classifier that creates a hyperplane using a group of features to separate states in the target class. SVM uses the Euclidean distance of the hyperplane from the nearest input values to determine the target state. A logistic regression model is fitted to the output of the support vector machine to obtain probability estimates. We used the default WEKA training error of 1.0E-12 and a default tolerance of the boundary of the hyperplane of 1.0E-03.

Random Forests [39]—This classifier generates predictive decision trees based on a random selection of features for creating every tree. A decision tree is a model that splits the training set into subsets based on the target class, until the splitting no longer adds value to the predictions. The final prediction is assigned by the consensus of voting by the individual trees. We randomly learned 1,000 trees with only one feature each. A large number of trees have been shown to increase the predictive accuracy of the RF [40].

2.5 Experimental Design

Figure 3 summarizes different experimental configurations in this study. We trained all machine learning classifiers using only the training dataset. Since some classifiers do not handle missing data, we used two approaches for training: 1) assigning all missing values to be "non-acute" (value F), and 2) assigning all missing values to be "missing" (value M). The first approach assumes that a finding with a missing value reported by Topaz implies that the finding is absent or non-acute in the patient, whereas the second approach does not make any assumptions and maps a missing finding to a "missing" value. BN models have the ability to deal with uncertainty in the data. To better understand the performance differences we used the three configurations for performance evaluation, according to the categories of missing information:

> *Configuration 1.* All missing values in both training and test datasets were assigned to value False.
>
> *Configuration 2.* All missing values in the training dataset were assigned to value False. All missing values in the test dataset were assigned to value Missing (M).
>
> *Configuration 3.* All missing values in the training dataset were assigned to value False. All missing values in the test dataset were left unassigned. This configuration is only possible to implement in Bayesian models.

The training dataset has a binary number of classes, either value T or value F. The case/control ratio is 468 : 29,004 (T:F), which is representative of the non-influenza season. Since there is an imbalance between the two classes, we performed additional experiments to train the classifiers under equal class ratio (468:468). We randomly resampled without replacement the controls in the training dataset and created 5 different datasets with the resampled controls and the full influenza cases. The process of dropping at random some cases from the majority class to give a balanced dataset is called Random Undersampling

(RUS) [41]. Then, we evaluated the average performance of the classifiers created under this condition using the test dataset.

We used two standard metrics for model evaluation: the area under the ROC curve (AUC) and the Brier Skill Score [42] (BSS). An ideal BSS is close to 1.0, while negative numbers indicate models that are less skilled than the weighted dice prediction of 0.0 (unskilled reference). The Brier Score (BS) in Equation 2 is measured as the average squared difference between the predicted value y_k and the observed value o_k, with the ideal score being 0 and the worst score being 1. On the other hand the Brier Skill Score (BSS) in Equation 3 is calculated as a scaled representation of the Brier Score relative to the relative frequency of the binary classes or reference Brier Score BS_{ref}.

$$BS=\frac{1}{n}\sum_{k=1}^{n}(y_k-o_k)^2 \quad (2)$$

$$BSS=1-\frac{BS}{BS_{ref}} \quad (3)$$

For example, BS_{ref} is equal to 0.098 in the test dataset with 9.8% of influenza cases, and let us assume that a hypothetical classification model has a BS of 0.25, then the BSS would be equal to BSS = 1 − (0.25/0.065) = −1.55, , which is considered an unskilled prediction. In this sense, it is better to use a BSS because it measures the difference between the score for the prediction and the score for the unskilled reference prediction, normalized by the total possible improvement that can be achieved. The ideal BSS score is1.

Measurements of the diagnostic tests are computed as ROC curves. The curve is constructed by varying the threshold to which the probability that is given by the classifier is considered of one class. In order to make comparisons between two curves, we used the nonparametric method developed by DeLong et al. [43], which is a commonly used method by biomedical researchers using the R package pROC [44]. This method computes correlation matrix between the curves, then it applies a χ^2 test to obtain a two-sided p-value of statistical difference between the curves. All experiments were run on a MacBook Pro with 2.7 GHz quad-core i7 processor and 8GB of RAM.

3 Results

In this section, we present the evaluation of the NLP tool under different acuteness values of influenza and the evaluation of classifiers under the configurations described in Section 2.5.

3.1 NLP Evaluation

Table 2 shows the results of an evaluation study demonstrating 90% accurate determination of 31 targeted findings of influenza in a test set of 122 reports for ED patients with influenza (by PCR test) between 9/1/2010 and 12/31/2011.

3.2 Classifiers Evaluation

Table 3 presents the classification results for the three configurations using the experimental design from Table 1. A tie between four algorithms (NB, LR, SVM, ANN) obtained the highest AUC (0.93). The expert-built model (*Expert*) in Configuration 3, where all missing values were left unassigned, obtained the lowest AUC (0.8). The expert-built model obtained the lowest AUC in all configurations. However, just by updating the priors for the expert-built model, to reflect that of the training dataset, the performance increased significantly (*Expert-MLE*). ANN obtained the highest BSS (0.41) in Configuration 2. *Expert-MLE* obtained the lowest BSS (−1.72) in Configuration 3. Bayesian classifiers are the only ones that can handle missing values without any preprocessing (Configuration 3). Nevertheless, the results from this configuration are not significantly different from Configuration 1, where missing values were assigned to False (F). Table 4 presents the classification performance results when a resampling technique was used. The classifiers trained with a balanced number of samples in each class achieve an equivalent classification performance than the same classifiers when not using resampling. All values in the table represent the averaged results of applying resampling without replacement 5 times to the training dataset.

4 Discussion

Effects of Missing Data in Classification

It is not uncommon to find ED reports with missing data. For example when there is no indication of fever noted in the ED report, the reason might be that the patient actually does not have fever, or the fever condition was not checked or reported, or the temperature was not yet assessed at the time of the report. In this study we assessed three different ways of dealing with missing values: missing at random (MAR) in Configuration 1, not missing at random (NMAR) in Configuration 2, and missing completely at random (MCAR) in Configuration 3. The previous study by Tsui et al. [13] adopted Configuration 3. Our results suggest that the missingness of information does not have an impact on the classification performance. We recommend the use of NMAR to deal with missing data. However, it does not deal with the issue of conflicting evidence among different reports and will be carefully analyzed in future work.

Effects of influenza Prevalence in Classification

Class imbalance has been extensively studied in the literature and it is found to be the source of classification over-fitting. However, given the volume of our datasets and the flexibility of our ML classifiers, we were able to achieve a high performance regardless of the prevalence rates in the training sets (50% or 1.6%). The 50% prevalence rate was achieved by resampling and the classifier trained by the high prevalence rate showed no statistically significant difference in classification performance with the classifier trained by the low prevalence rate. We attribute this behavior to the large number of ED reports used that created a well-trained classifier when testing under the hypothetical influenza outbreak prevalence (9.8%). Given that electronic ED reports are increasingly being used in many health systems, we recommend the use of large datasets for training of similar classifiers.

Classifiers performance

The selection of machine learning classifiers includes Bayesian classifiers (NB, K2-BN), function classifiers (LR, ANN, SVM), and decision trees (RF), which are commonly used in the literature. EBMC is a novel Bayesian classifier that has shown promising results in genomic datasets. We used the performance metric of AUC, which has a maximum value of one indicating that for all patients the classifier was able to correctly detect the status of the disease. A value of 0.5 or less would indicate that the classifier was not better than random guessing among classes. Overall, the results in each configuration were similar between all classifiers (no statistically significant difference). Such results are not surprising given the large number of training cases and the capabilities of the classifiers.

However, using AUC alone for classifier evaluation may be biased by the prevalence of test data. For example, a hypothetical classifier that makes predictions in a dataset with low (influenza) prevalence rate (e.g., 10% influenza present, 90% influenza absent) would achieve an AUC higher than 0.5. This is called the unskilled classifier problem, because it refers to a classifier that performs better than random without any training effort. To address this issue we calculated the Brier Skill Score (BSS), which is a measure of calibration. The BSS creates an index between −1 and 1 that provides information as of how far away the results of any classifier are in relation to the unskilled classifier. We can infer from our results that for the most part the expert model is unskilled due to negative BSS scores. In contrast, all ML classifiers achieve positive BSS scores, which indicate that all of them have the ability to perform better than the unskilled classifier.

Use of Naïve Bayes and Logistic Regression

It has been suggested that the predictions obtained from a LR model are the same as those predictions originated from a NB model [46]. LR is consistent with the conditional independence assumption used in NB. Nevertheless, there are important differences in each algorithm. LR will adjust its parameters to maximize the conditional likelihood of the data, even if the resulting parameters are inconsistent with the NB parameter estimates [47]. Furthermore, the possibility of making predictions in the presence of missing data is a characteristic that is better modeled in a Bayesian approach. The NB model has prior parameter estimates that are obtained during the training step, and the prediction of a new case can be done without imputing any missing data.

Limitations

Our study had several limitations. 1) The study data came from a single health system. 2) Our method used only one NLP tool (Topaz); however, it was compared with MedLEE in our previous study [14] and demonstrated similar performance with MedLEE. 3) We only used a small subset of UMLS codes instead of extracting all available clinical concepts and applying feature selection methods to identify risk factors for each classifier. 4) The expert-built prediction model and the selection of 31 influenza attributes may be biased.

5 Conclusion

This study demonstrated that 1) ML classifiers had a better performance than expert constructed classifiers given a particular NLP extraction system, 2) using a large number of ED notes allowed machine learning classifiers to automatically build models that can detect influenza cases, 3) missing clinical information marked as a value of missing (not missing at random) had a consistently improved performance among 3 (out of 4) ML classifiers when it was compared with the configuration of not assigning a value of missing (missing completely at random). 4) Given a large number of training cases the class imbalance problem does not affect the classification performance. Since the meaningful use promotes the use of electronic health records (EHR) for all hospitals in the United States [48], analysis of this data could play an important role in public health surveillance of various diseases. This study suggests that analyzing information from the EHR using machine learning classifiers can achieve significant accuracies in the presence of abundant clinical reports.

Acknowledgments

We thank Mr. Hoah-Der Su for preparing the datasets that we used in the experiments.

This research was funded by grants R01LM011370-01A1 and R01LM010020 from the NLM, grant P01HK000086 from the CDC in support of the University of Pittsburgh Center for Advanced Study of Public Health in Informatics, grant U38HK000063 from CDC, and grant SAP 40000012020 from the Pennsylvania Department of Health, and grant IIS-0911032 from the NSF. The International Fulbright S&T Award and CONACyT-Mexico supported ALP.

References

1. Bradley CA, Rolka H, Walker D, Loonsk J. BioSense: Implementation of a National Early Event Detection and Situational Awareness System. MMWR Morb Mortal Wkly Rep. 2005
2. Wagner, MM.; Moore, AW.; Aryel, RM. Handbook of Biosurveillance. Academic Press; 2011.
3. Reina J, Plasencia V, Leyes M, Nicolau A, Galmés A, Arbona G. Comparison study of a real-time reverse transcription polymerase chain reaction assay with an enzyme immunoassay and shell vial culture for influenza A and B virus detection in adult patients. Enferm Infecc Microbiol Clin. 2010; 28:95–8.10.1016/j.eimc.2008.11.021 [PubMed: 19477042]
4. Tsui F-C, Espino JU, Sriburadej T, Su H, Dowling JN. Building an automated Bayesian case detection system. Emerging Health Threats Journal. 2011:68–9.10.3134/ehtj.10.101
5. Shu B, Wu K-H, Emery S, Villanueva J, Johnson R, Guthrie E, et al. Design and Performance of the CDC Real-Time Reverse Transcriptase PCR Swine Flu Panel for Detection of 2009 A (H1N1) Pandemic Influenza Virus. J Clin Microbiol. 2011; 49:2614–9.10.1128/JCM.02636-10 [PubMed: 21593260]
6. Hurt AC, Alexander R, Hibbert J, Deed N, Barr IG. Performance of six influenza rapid tests in detecting human influenza in clinical specimens. Journal of Clinical Virology. 2007; 39:132–5.10.1016/j.jcv.2007.03.002 [PubMed: 17452000]
7. Espy MJ, Uhl JR, Sloan LM, Buckwalter SP, Jones MF, Vetter EA, et al. Real-Time PCR in Clinical Microbiology: Applications for Routine Laboratory Testing. Clin Microbiol Rev. 2006; 19:165–256.10.1128/CMR.19.1.165-256.2006 [PubMed: 16418529]
8. Echevarría-Zuno S, Mejía-Aranguré JM, Mar-Obeso AJ, Grajales-Muñiz C, Robles-Pérez E, González-León M, et al. Infection and death from influenza A H1N1 virus in Mexico:a retrospective analysis. The Lancet. 2009; 374:2072–9.10.1016/S0140-6736(09)61638-X
9. Nachtnebel M, Greutelaers B, Falkenhorst G, Jorgensen P, Dehnert M, Schweiger B, et al. Lessons from a one-year hospital-based surveillance of acute respiratory infections in Berlin- comparing case definitions to monitor influenza. BMC Public Health. 2012; 12:245.10.1186/1471-2458-12-245 [PubMed: 22452874]

10. May LSL, Griffin BAB, Bauers NMN, Jain AA, Mitchum MM, Sikka NN, et al. Emergency department chief complaint and diagnosis data to detect influenza-like illness with an electronic medical record. CORD Conference Proceedings. 2010; 11:1–9.

11. Tsui F-C, Wagner MM, Dato V, Chang C-CH. Value of ICD-9-Coded Chief Complaints for Detection of Epidemics. J Am Med Inform Assoc. 2002; 9:S41–7.10.1197/jamia.M1224

12. Elkin PL, Froehling DA, Wahner-Roedler DL, Brown SH, Bailey KR. Comparison of natural language processing biosurveillance methods for identifying influenza from encounter notes. Ann Intern Med. 2012; 156:11–8.10.7326/0003-4819-156-1-201201030-00003 [PubMed: 22213490]

13. Tsui F-C, Wagner M, Cooper GF, Que J, Harkema H, Dowling JN, et al. Probabilistic case detection for disease surveillance using data in electronic medical records. Online J Public Health Inform. 2011; 310.5210/ojphi.v3i3.3793

14. Ye Y, Tsui F-C, Wagner M, Espino JU, Li Q. Influenza detection from emergency department reports using natural language processing and Bayesian network classifiers. J Am Med Inform Assoc. 2014 amiajnl–2013–001934. 10.1136/amiajnl-2013-001934

15. Lopez Pineda A, Tsui F-C, Visweswaran S, Cooper GF. Detection of Patients with Influenza Syndrome Using Machine-Learning Models Learned from Emergency Department Reports. Online J Public Health Inform. 2013; 510.5210/ojphi.v5i1.4446

16. Lin J-H, Haug PJ. Exploiting missing clinical data in Bayesian network modeling for predicting medical problems. Journal of Biomedical Informatics. 2008; 41:1–14.10.1016/j.jbi.2007.06.001 [PubMed: 17625974]

17. Chapman WW, Conway M, Dowling JN. Challenges in adapting an natural language processing system for real-time surveillance. Using Information in …. 2011

18. Chapman WW, Harkema H. C-C1-03: Identifying Respiratory-Related Clinical Conditions From ED Reports With Topaz. Clin Med Res. 2010; 8:53–3.10.3121/cmr.8.1.53-b

19. Zou Q, Chu WW, Morioka C, Leazer GH, Kangarloo H. IndexFinder: a method of extracting key concepts from clinical texts for indexing. AMIA Annu Symp Proc. 2003:763–7. [PubMed: 14728276]

20. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004; 32:D267–70.10.1093/nar/gkh061 [PubMed: 14681409]

21. Conway, M.; Dowling, JN.; Chapman, WW. Developing a Biosurveillance Application Ontology for Influenza-Like-Illness. th Workshop on Ontologies and Lexical Resources Ontolex; Beijing. 2010; p. 58-66.

22. Conway, M.; Dowling, JN.; Chapman, WW. Developing an Application Ontology for Mining Free Text Clinical Reports: The Extended Syndromic Surveillance Ontology. Third International Workshop on Health Document Text Mining and Information Analysis; LOUHI. 2011; p. 75-82.

23. Harkema H, Dowling JN, Thornblade T, Chapman WW. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. Journal of Biomedical Informatics. 2009; 42:839–51.10.1016/j.jbi.2009.05.002 [PubMed: 19435614]

24. Chapman WW, Dowling JN, Wagner MM. Classification of Emergency Department Chief Complaints Into 7 Syndromes: A Retrospective Analysis of 527,228 Patients. Annals of Emergency Medicine. 2005; 46:445–55.10.1016/j.annemergmed.2005.04.012 [PubMed: 16271676]

25. Buchanan, BG.; Chapman, WW.; Cooper, GF.; Hanbury, P.; Kayaalp, M.; Ramachandran, M., et al. Creating a Software Tool for the Clinical Researcher -- the IPS System. Proceedings of the AMIA Symposium; 2002; p. 1210

26. Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc. 2004; 11:392–402.10.1197/jamia.M1552 [PubMed: 15187068]

27. Reed C, Angulo FJ, Swerdlow DL, Lipsitch M, Meltzer MI, Jernigan D, et al. Estimates of the prevalence of pandemic (H1N1) 2009, United States, April-July 2009. Emerging Infect Dis. 2009; 15:2004–7.10.3201/eid1512.091413 [PubMed: 19961687]

28. Druzdzel MJ. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a development environment for graphical decision-theoretic models. Aaai/Iaai. 1999

29. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. SIGKDD Explor Newsl. 2009; 11:10.10.1145/1656274.1656278
30. Neapolitan, RE. Probabilistic Reasoning in Expert Systems. 2012.
31. John, GH.; Langley, P. Estimating continuous distributions in Bayesian classifiers. Morgan Kaufmann Publishers Inc; 1995.
32. Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Mach Learn. 1992; 9:309–47.10.1007/BF00994110
33. Cooper GF, Hennings-Yeomans P, Visweswaran S, Barmada M. An efficient bayesian method for predicting clinical outcomes from genome-wide data. AMIA Annu Symp Proc. 2010; 2010:127–31. [PubMed: 21346954]
34. Jiang X, Cai B, Xue D, Lu X, Cooper GF, Neapolitan RE. A comparative analysis of methods for predicting clinical outcomes using high-dimensional genomic datasets. J Am Med Inform Assoc. 2014; 21:e312–9.10.1136/amiajnl-2013-002358 [PubMed: 24737607]
35. Hoerl AE, Kennard RW. Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics. 2012; 12:55–67.10.1080/00401706.1970.10488634
36. Rumelhart, DE.; Hinton, GE.; Williams, RJ. Learning Internal Representations by Error Propagation. 1985.
37. Ghaffari A, Abdollahi H, Khoshayand MR, Bozchalooi IS, Dadgar A, Rafiee-Tehrani M. Performance comparison of neural network training algorithms in modeling of bimodal drug delivery. International Journal of Pharmaceutics. 2006; 327:126–38.10.1016/j.ijpharm.2006.07.056 [PubMed: 16959449]
38. Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Research. 1998
39. Breiman L. Random Forests. Mach Learn. 2001; 45:5–32.10.1023/A:1010933404324
40. Khoshgoftaar, TM.; Golawala, M.; Van Hulse, J. An Empirical Study of Learning from Imbalanced Data Using Random Forest. Vol. 2. IEEE; 2007. p. 310-7.
41. Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intelligent Data Analysis. 2002; 6:429–49.
42. Wilks, DS. Statistical Methods in the Atmospheric Sciences. Academic Press; 2011.
43. DeLong ERE, DeLong DMD, Clarke-Pearson DLD. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. Biometrics. 1988; 44:837–45.10.2307/2531595 [PubMed: 3203132]
44. Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez J-C, et al. pROC: an open-source package for R and S plus to analyze and compare ROC curves. BMC Bioinformatics. 2011; 12:77.10.1186/1471-2105-12-77 [PubMed: 21414208]
45. Efron, B.; Tibshirani, RJ. An Introduction to the Bootstrap. CRC Press; 1994.
46. Sebastiani P, Solovieff N, Sun JX. Naïve Bayesian Classifier and Genetic Risk Score for Genetic Risk Prediction of a Categorical Trait: Not so Different after all! Front Genet. 2012; 3:26–6.10.3389/fgene.2012.00026 [PubMed: 22393331]
47. Mitchell TM. Generative and Discriminative Classifiers: Naïve Bayes and Logistic Regression. Machine Learning. 2015:1–17.
48. Sittig DF, Singh H. Electronic Health Records and National Patient-Safety Goals. N Engl J Med. 2012; 367:1854–60.10.1056/NEJMsb1205420 [PubMed: 23134389]

Highlights

1. ML classifiers performed better than an expert constructed classifier
2. Influenza cases can be detected with ML classifiers built from abundant ED reports processed by a natural language processing tool
3. Missing clinical data marked as a value *missing* improved the ML classifiers performance
4. ML classifiers performance was not affected by class imbalance, given abundant training samples

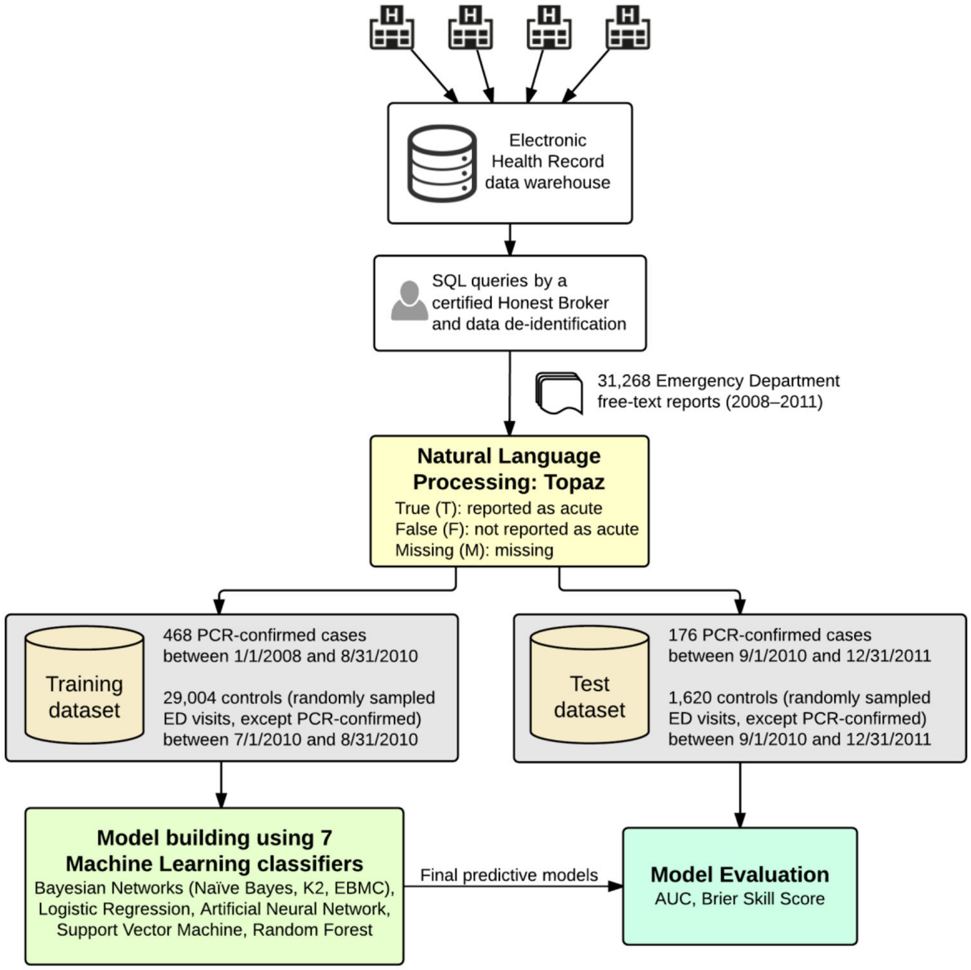


Figure 1.

Study process flow.

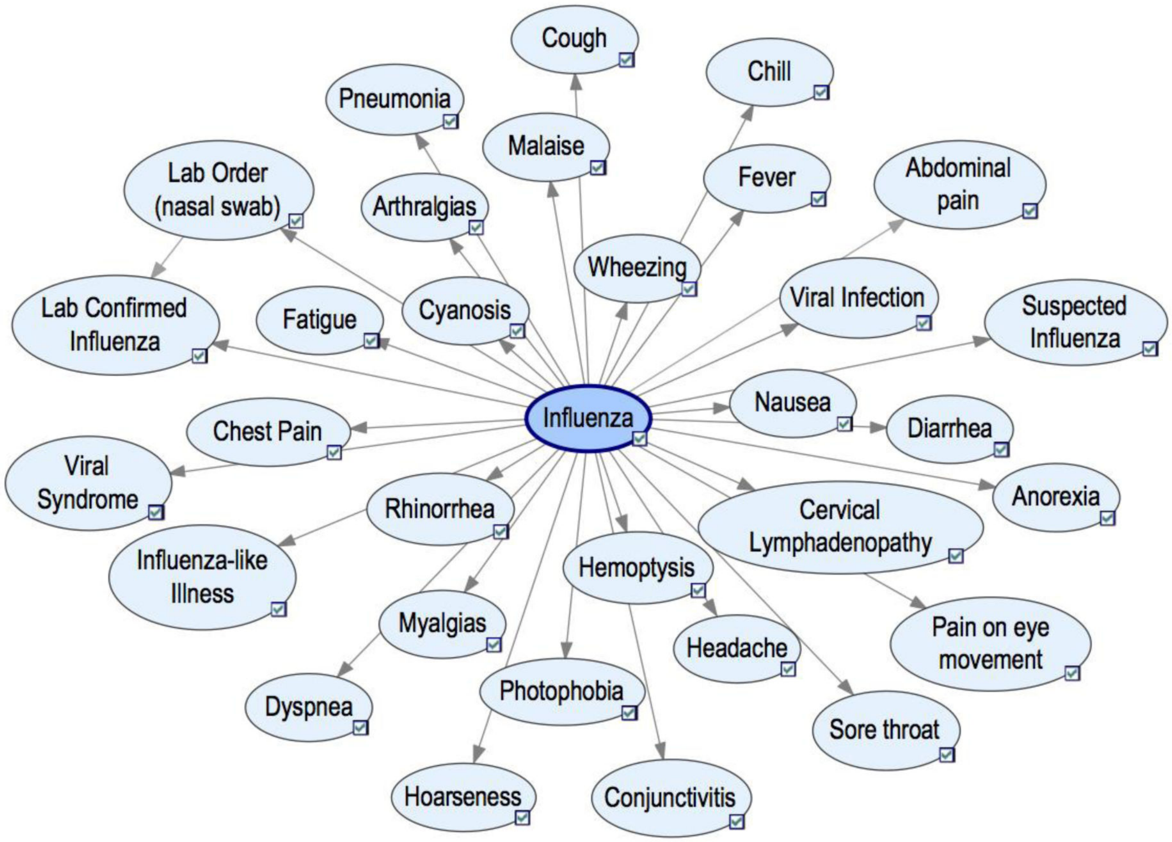


Figure 2.
Expert-constructed Bayesian model.

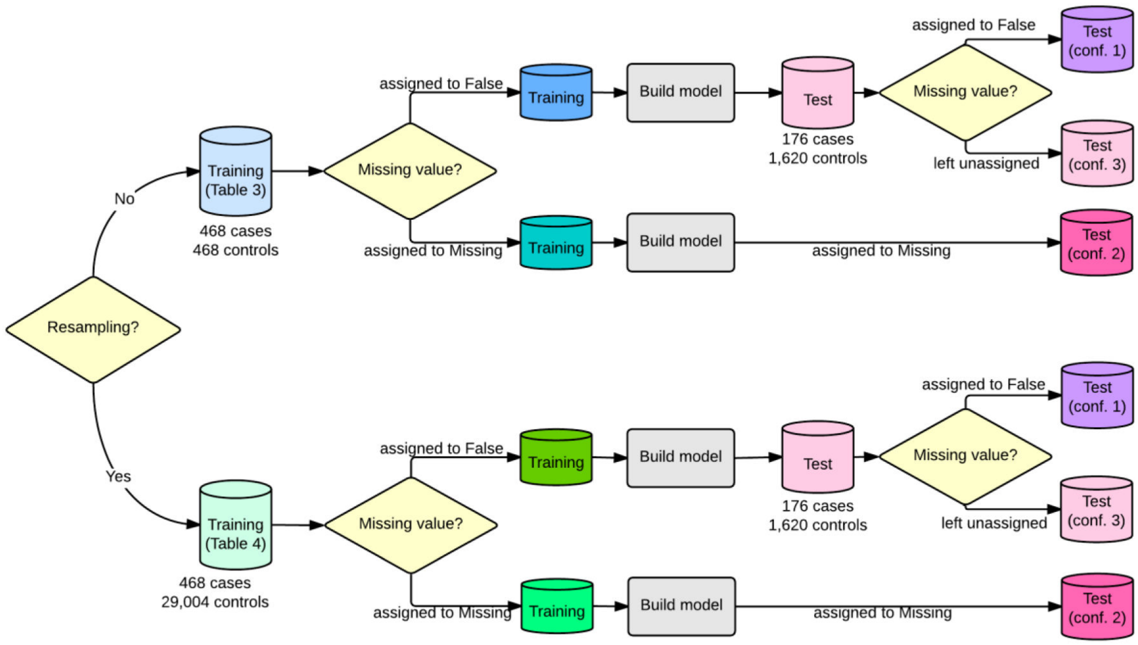


Figure 3.
Experiments tree.

Table 1

Summary of the configurations and experimental design

#	UMLS	Description
1	C0420679	Nasal swab taken
2	LC0021400*	NLP - Lab confirmed flu
3	C0521839	Influenza-like Illness
4	IC0021400*	Suspected Flu
5	C0042740	Viral Syndrome
6	C0231528	Myalgias
7	C1260880	Rhinorrhea
8	C0521026	Viral
9	C0015967	Fever
10	C0010200	Coughing
11	C0085593	Chills
12	C0242429	Sore Throat
13	C0231218	Malaise
14	C0003862	Arthralgia
15	C0032285	Pneumonia
16	C0043144	Wheezing
17	C0019825	Hoarseness
18	C0235592	Cervical lymphadenopathy
19	C0018681	Headache
20	C0019079	Hemoptypsis
21	C0015672	Fatigue
22	C0011991	Diarrhea
23	C0009763	Conjunctivitis
24	C0013404	Dyspnea
25	C0003123	Anorexia
26	C0027497	Nausea
27	C0008031	Chest Pain
28	C0010520	Cyanosis
29	C0239430	Pain with eye movement
30	C0085636	Photophobia
31	C0000729	Abdominal Cramps

*Topaz used two non-UMLS codes LC0021400 and IC0021400 to represent laboratory-tested influenza and suspected influenza extracted from free-text ED reports, respectively. Note that the two findings were not defined in 2008 version of UMLS codes.

Table 2

NLP performance for 31 influenza findings.

Type of Finding	Statistic	Result (C.I.*)
non-acute	Precision	0.90 (0.87, 0.92)
	Recall	0.79 (0.75, 0.82)
acute	Precision	0.86 (0.83, 0.89)
	Recall	0.78 (0.75, 0.81)
acute + non-acute	Precision	0.94 (0.93, 0.95)
	Recall	0.84 (0.82, 0.86)
	Accuracy	0.78 (0.76, 0.81)
acute + non-acute + not mentioned	Accuracy	0.90 (0.89, 0.91)
	Kappa between clinician annotation and Topaz findings	0.81 (0.79, 0.83)

*The 95% confidence interval of the empirical distribution was obtained by bootstrapping with replacement (2000 times), and it was calculated using bootstrap percentiles [45] with SAS® software 9.3

Table 3

Evaluation results without resampling. The statistical difference is compared to the best performing classifier under each configuration. Only Bayesian classifiers are used in Configuration 3 because they can handle missing data natively.

Conf.	Algorithm	AUC	95% C.I.	Statistical Difference to Best Performing Algorithm p-value	Brier Skill Score	Train Time (seconds)
1	NB	0.93	(0.91, 0.95)	1.0	0.35	0.05
1	LR	0.93	(0.91, 0.95)	0.75	0.39	1.71
1	ANN	0.93	(0.9, 0.95)	0.54	0.38	269.20
1	SVM	0.92	(0.9, 0.95)	0.39	0.11	16.73
1	K2-BN	0.92	(0.89, 0.94)	0.09	0.37	3.07
1	EBMC	0.91	(0.88, 0.94)	0.01	0.27	9.41
1	Expert-MLE	0.88	(0.85, 0.91)	< 0.001	−0.66	Manual + 0.12
1	RF	0.87	(0.83, 0.91)	< 0.001	0.33	162.80
1	Expert	0.87	(0.84, 0.9)	< 0.001	0	Manual
2	NB	0.93	(0.91, 0.95)	1.0	0.34	0.03
2	Expert-MLE	0.93	(0.91, 0.95)	0.25	0.34	Manual + 0.08
2	LR	0.92	(0.89, 0.94)	0.26	0.4	3.77
2	RF	0.92	(0.9, 0.94)	0.22	0.25	60.89
2	SVM	0.9	(0.87, 0.93)	0.03	0.38	108.05
2	ANN	0.91	(0.88, 0.94)	0.07	0.41	1902.40
2	K2-BN	0.89	(0.86, 0.92)	0.004	0.36	8.41
2	EBMC	0.84	(0.81, 0.87)	< 0.001	0.38	10.06
3	NB	0.92	(0.89, 0.95)	1.0	0.22	0.05
3	K2-BN	0.9	(0.88, 0.93)	0.02	0.4	3.07
3	EBMC	0.89	(0.86, 0.92)	0.001	0.38	9.41
3	Expert-MLE	0.88	(0.85, 0.91)	< 0.001	−1.72	Manual + 0.12
3	Expert	0.8	(0.77, 0.84)	< 0.001	−0.35	Manual

Table 4

Evaluation results with resampling. This table presents the results of applying resampling to the training dataset to obtain the same number of cases and controls. All values represent the averaged results of applying resampling without replacement 5 times to the training dataset. It also shows the statistical difference between the AUC performances of the models using resampling and the results from Table 3, which does not use resampling.

Conf.	Algorithm	AUC	95% C.I.	Statistical difference between resampling and no-resampling p-value	Brier Skill Score
1	NB	0.93	(0.91, 0.95)	0.11	0.08
1	LR	0.93	(0.90, 0.95)	0.58	0.11
1	SVM	0.93	(0.90, 0.95)	0.42	0.09
1	K2-BN	0.92	(0.90, 0.94)	0.71	0.03
1	RF	0.92	(0.90, 0.94)	< 0.001	0.01
1	ANN	0.92	(0.89, 0.94)	0.43	0.04
1	EBMC	0.91	(0.88, 0.93)	0.31	0.24
1	Expert-MLE	0.89	(0.86, 0.91)	0.15	0.02
2	NB	0.93	(0.90, 0.95)	0.21	0.07
2	Expert-MLE	0.93	(0.90, 0.95)	0.2	0.06
2	RF	0.92	(0.90, 0.95)	0.44	−0.02
2	SVM	0.92	(0.89, 0.94)	0.16	0.01
2	LR	0.92	(0.89, 0.94)	0.55	−0.08
2	K2-BN	0.92	(0.89, 0.94)	0.01	−0.05
2	EBMC	0.91	(0.89, 0.94)	0.002	0.04
2	ANN	0.91	(0.88, 0.94)	0.5	−0.08
3	NB	0.92	(0.89, 0.94)	0.74	−1.58
3	K2-BN	0.91	(0.88, 0.93)	0.3	−1.4
3	EBMC	0.89	(0.86, 0.92)	0.24	−1.38
3	Expert-MLE	0.87	(0.84, 0.91)	0.56	−2.04