Published in final edited form as:
Int J Med Inform. 2011 January ; 80(1): 56–66. doi:10.1016/j.ijmedinf.2010.10.015.

An exploratory study of a text classification framework for Internet-based surveillance of emerging epidemics

Manabu Torii[a,*], **Lanlan Yin**[b], **Thang Nguyen**[a,1], **Chand T. Mazumdar**[a], **Hongfang Liu**[b], **David M. Hartley**[a,b,c,2], and **Noele P. Nelson**[a,d,2]

[a]The ISIS Center, Georgetown University Medical Center, Washington, DC, USA

[b]Department of Biostatistics, Bioinformatics and Biomathematics, Georgetown University Medical Center, Washington, DC, USA

[c]Departments of Microbiology and Immunology and Radiology, Georgetown University Medical Center, Washington, DC, USA

[d]Department of Pediatrics, Georgetown University Medical Center, Washington, DC, USA

Abstract

Purpose—Early detection of infectious disease outbreaks is crucial to protecting the public health of a society. Online news articles provide timely information on disease outbreaks worldwide. In this study, we investigated automated detection of articles relevant to disease outbreaks using machine learning classifiers. In a real-life setting, it is expensive to prepare a training data set for classifiers, which usually consists of manually labeled relevant and irrelevant articles. To mitigate this challenge, we examined the use of randomly sampled unlabeled articles as well as labeled relevant articles.

Methods—Naïve Bayes and Support Vector Machine (SVM) classifiers were trained on 149 relevant and 149 or more randomly sampled unlabeled articles. Diverse classifiers were trained by varying the number of sampled unlabeled articles and also the number of word features. The trained classifiers were applied to 15 thousand articles published over 15 days. Top-ranked articles from each classifier were pooled and the resulting set of 1337 articles was reviewed by an expert analyst to evaluate the classifiers.

Results—Daily averages of areas under ROC curves (AUCs) over the 15-day evaluation period were 0.841 and 0.836, respectively, for the naïve Bayes and SVM classifier. We referenced a database of disease outbreak reports to confirm that this evaluation data set resulted from the pooling method indeed covered incidents recorded in the database during the evaluation period.

Conclusions—The proposed text classification framework utilizing randomly sampled unlabeled articles can facilitate a cost-effective approach to training machine learning classifiers in

[*]Corresponding author at: ISIS Center, Georgetown University Medical Center, 2115 Wisconsin Avenue, NW, Suite 603, Washington, DC 20057, USA. Tel.: +1 202 687 8520; fax: +1 202 687 9339. torii@isis.georgetown.edu (M. Torii).
[1]T.N. is currently affiliated with National Library of Medicine, National Institute of Health, Bethesda, MD 20894, USA.
[2]These authors contributed equally to this work.

Authors' contributions

MT designed the experiments and developed the programs with LY, TN and HL CTM contributed to the data acquisition and the system evaluation. DMH and NPN coordinated the resources to execute this study. All the authors contributed significantly to the analysis and interpretation of the experimental results. MT drafted this manuscript and the other authors critically revised it.

Conflict of interest

The authors declare no conflict of interest.

a real-life Internet-based biosurveillance project. We plan to examine this framework further using larger data sets and using articles in non-English languages.

Keywords

Natural language processing; Information storage and retrieval; Medical informatics applications; Disease notification; Disease outbreaks; Biosurveillance; Internet

1. Introduction

Textual information sources on the Internet, such as publically available news media and newsletters, have been found informative for early detection of emerging epidemics. Over the years, several Internet-focused, event-based biosurveillance projects have been launched to mine disease outbreak information from online articles [1–7]. For example, the Global Public Health Intelligence Network (GPHIN), an early warning system developed by the Public Health Agency of Canada, mines information pertaining to public health events from Internet media sources. In November 2002 GPHIN identified unusual disease cases reported in local news articles in China [8], which the World Health Organization (WHO) recognizes as one of the earliest alerts of the 2002–2003 SARS outbreak [9]. Project Argus is another Internet-based biosurveillance project hosted at the Georgetown University Medical Center. A retrospective study of the SARS outbreak by Georgetown researchers showed that subtle indications (unseasonal bad influenza) were reported in online news sources as early as September 2002 [10]. Despite the great potential in mining early indications of disease outbreaks from online sources, identification of relevant online articles is challenging in practice due to the vast amount of ever growing publications on the Internet. Automated identification of relevant articles is the first step toward effective Internet-based biosurveillance.

Machine learning classifiers such as naïve Bayes and Support Vector Machine (SVM) classifiers are effective approaches to detecting online articles pertaining to infectious disease outbreaks in Internet-based biosurveillance projects [11–14]. The development of a machine learning text classifier, however, requires a large set of class-labeled articles (a training corpus) representative of those in the application domain. In a time-sensitive biosurveillance project, manual labeling of archived articles is inefficient and expensive, because analysts must work on the labeling task outside of the normal operational hours. Moreover, their classification might become obsolete, since disease and event types targeted in the project may change over time. As an alternative to labeling archived articles, analysts may be requested to label articles as part of their surveillance work. Given simple classes of articles such as “relevant” and “irrelevant” (to the surveillance topic), real-time labeling of article classes does not interfere with the regular surveillance workflow, and analysis of source articles in this manner is an important part of Internet-based surveillance. Moreover, this approach has the advantage of enabling maintenance of an up-to-date training corpus in this application domain where the distribution of articles constantly and rapidly changes. Meanwhile, a corpus created in such a framework would be a biased set because analysts strive to search for only relevant articles. In this framework, therefore, it is difficult to assemble a corpus representative of articles in the domain, which should include not only relevant articles but also diverse irrelevant articles.

In this study, we examine an approach to compiling a training corpus in real-time surveillance, and build a machine learning text classifier in a non-intrusive manner for analysts. Specifically, we focus on English articles published by selected sources in South Asia, and evaluate machine learning text classifiers trained on manually identified relevant articles and unlabeled articles sampled as likely irrelevant articles. In the following sections,

we introduce the background of the study, including an overview of Project Argus, and discuss the machine learning methods. Next, we present the experimental methods, followed by the results and the discussion of the experiments. We then summarize our findings and address future work.

2. Background

2.1. Project Argus

Project Argus is a biosurveillance project which uses publically available news media to provide early warning and tracking of emerging biological threats [7,15,16]. The project involves a team of analysts who monitor information sources in about 40 languages. To search articles relevant to the surveillance topic, analysts use a customized information retrieval system, where the effective Boolean query strings have been developed and refined over time for different topics. Despite the great potential of automated text classifiers to help analysts identify relevant articles, it is challenging in practice to develop and maintain an up-to-date class-labeled corpus required to train/adjust classifiers in this constantly evolving application domain.

2.2. Machine learning text classifiers

Text categorization has been an active field of research that has been applied to various real-world applications, e.g., spam email filtering [17], biological database curation [18,19], syndromic surveillance [20], and Internet-based biosurveillance [11–13,21], among many others. In building a supervised machine learning text classifier, a sufficiently large class-labeled corpus is needed to serve as the training corpus. Each article in the training corpus is converted into a feature vector (e.g., weighted frequencies of pre-selected words in the article), and then a machine learning algorithm can be applied to build a classification model (classifier).

In the following subsections, we review two machine learning algorithms widely used for text classification, naïve Bayes and SVM, which we also used in this study. We then discuss feature selection methods.

2.2.1. Machine learning text classifiers—Among other machine learning algorithms, naïve Bayes and SVM are two of the most widely used text classification algorithms. Naïve Bayes classifiers are simple but effective in practice [22], and they have already been used for Internet-based biosurveillance [12,13]. A naïve Bayes classifier calculates the probability of an article belonging to a particular document class, $P(\text{class} \mid \text{article}) = P(\text{class}) \times P(\text{article} \mid \text{class})/P(\text{article})$, and it can be used to prioritize articles relevant to a particular topic. Namely, articles can be sorted on the predicted probability of an article belonging to the "relevant" class, $P(\text{class}=\text{"relevant"} \mid \text{article})$. The multinomial naïve Bayes model is one type of naïve Bayes model suitable for text classification [23]. In this model, assuming the conditional independence of word occurrences in articles, $P(\text{class} \mid \text{article})$ is calculated based on $P(\text{class}, \text{article}) = P(\text{class}) \prod_w P(w/\text{class})$, where w is a word (i.e., feature) found in the article, and $P(\text{class})$ and $P(w \mid \text{class})$ are probabilities derived from the training corpus.[3] Despite their good overall classification results, probabilities estimated by naïve Bayes classifiers are not well-calibrated [24].

[3]To avoid $P(w \mid \text{class}) = 0$ that makes $P(\text{class} \mid \text{article}) = 0$, *smoothing* is used. Add-one smoothing is a simple method, in which all the selected feature words are assumed to appear in all the different classes at least once.

SVM [25] is a powerful machine learning paradigm that has been often reported to outperform other machine learning classifiers [26]. SVM classifiers can exploit a large number of features while avoiding over-fitting to the training data. Given two sets of instances belonging to two classes, SVM seeks a hyper-plane in the feature space that maximizes the *margin* between the two sets of instances. When instances are not linearly separable or a large margin is attainable by overlooking (misclassifying) some instances, the soft-margin method can be used to allow misclassification at a defined cost for each misclassified instance. A non-linear SVM classifier can be built, but past studies suggest that a linear SVM classifier is usually sufficient for text data [27]. The derivation of a hyper-plane is a numerical optimization process that can be computationally expensive, but efficient implementations of SVM have been publicly available, e.g., LibSVM and SVM^{light}. An output from a SVM classifier reflects the distance of an instance from a derived hyper-plane, but calibration of probabilities for predicted classes has also been studied for SVM [28,29].

2.2.2. Feature selection—Features commonly used for text classifiers are words. A powerful machine learning algorithm such as SVM can accommodate a large number of word features without over-fitting to the training data, but they may perform better with a selected subset of available words in terms of computational costs and/or classification performance [30,31]. Information gain (IG) is a widely used feature selection method. Given a class-labeled article set, A, let A_w be a subset of A that contains articles with a feature word, w. Then IG can be calculated as below to measure the utility of the word w for article classification:

$$\mathrm{IG}(A,w)=H(A)-\left(\frac{|A_w|}{|A|}H(A_w)+\frac{|A|-|A_w|}{|A|}H(A-A_w)\right).$$

Here, $|S|$ is the number of instances in an article set S, and $H(S)$ is the entropy of S calculated by $H(S)=-\sum_{\mathrm{class}}P_S(\mathrm{class})\log_2 P_s(\mathrm{class})$, where $P_S(\mathrm{class})$ is the ratio of instances (articles) that belong to the specified class in the set S. Entropy is an index of the uncertainty in predicting classes, and IG measures the reduction in entropy after knowing the presence or absence of a word in an article. Therefore, in general, as the IG increases, the utility of the word for article classification also increases. There are other methods to measure the utility of a feature, such as χ^2 statistics or point-wise mutual information. In our preliminary experiments, classifiers based on IG were as good as or better than those derived using the other feature selection methods, and we used IG for this work. With these measures, features can be sorted according to their utility, and a selected number of words at the top of the sorted list are used by classifiers.

3. Experiments

We examined a non-intrusive framework for analysts to compile a training corpus of machine learning classifiers in Internet-based biosurveillance. We trained naïve Bayes and SVM classifiers on a corpus compiled in real-time surveillance of Project Argus, and evaluated the classification performance through manual review by an Argus analyst. Note, however, that the evaluation conducted in the current study (see Fig. 2) is not part of the proposed framework, which requires analysts to thoroughly review classification results outside of their regular workflow. Fig. 1 shows the overview of the proposed framework. The proposed framework allows for an automated filtering of articles (Fig. 1(a) → (d) → (c)), where the text classification system is built on a corpus readily compiled in the regular

surveillance workflow (Fig. 1(e)), and serves as an alternative to manual retrieval and filtering of articles (Fig. 1(a) → (b) → (c)).

3.1. Training corpus

We selected target geographic regions in South Asia, and requested an Argus regional analyst to manually classify articles during the real-time surveillance work. After one month, we compiled a set of 149 relevant articles (positive instances). While these articles were reviewed by only one analyst, it could be reasonably believed that they were truly relevant since the experienced analyst had read and explicitly labeled them (see also Section 4 on inter-coder agreement). To compile a set of irrelevant articles (negative instances), during the same time period, we retrieved 40 thousand articles published by the same online sources using broad Boolean search strings. For the majority of these articles, only the title was reviewed by the analyst during the real-time surveillance, and selected articles were read in detail to be labeled. We assumed that the articles that were not explicitly labeled as relevant (positive) were irrelevant (negative). This assumption could be at fault, for example, when multiple articles reported the same incident, and analysts did not label every such article. Of the 40 thousand articles retrieved, however, only a small fraction could be relevant in practice, especially after articles with (almost) the same titles were checked as described. Therefore, our assumption of unlabeled articles being irrelevant generally holds. Moreover, Noto et al. showed that a classifier trained to prioritize labeled (relevant) articles over unlabeled ones also prioritizes relevant articles over irrelevant ones under the condition that labeled articles are random selections among relevant articles [19]. Assuming that unlabeled relevant articles, if any, are primarily (near-) duplicates of labeled relevant articles and also that analysts chose to read/label articles without a systematic bias, the problem setting assumed by Noto et al. likely holds here. In other words, even if we train classifiers on labeled relevant and unlabeled articles (instead of labeled relevant and labeled irrelevant articles), the resulting classifiers can prioritize/rank articles as intended with reasonable performance.

3.2. Evaluation framework

We used a *pooling* method to evaluate classifiers built on the training corpus (Fig. 2) [32] because it was not feasible for analysts to review and manually label all 40 thousand articles. Among the articles retrieved over 15 days following the one month period of the training corpus collection, we compiled a set of high-priority articles predicted by diverse classifiers built on the training corpus. Then, articles in this set were manually classified by an analyst in order to evaluate the classifiers. This evaluation method will not examine the true sensitivity of classifiers because there can be relevant articles outside of the predicted high-priority articles. To gain insights into the true sensitivity, apart from the evaluation using the pooling method, we manually examined if incidents detected by Argus analysts during the real-time operation could be found among high-priority articles identified by automated classifiers (Section 4.2). We used receiver operating characteristic (ROC) curves and areas under ROC curves (AUC) to evaluate classifiers [33].

3.3. Machine learning text classifiers

LibSVM implementation of SVM [28] and Weka implementation of multinomial naïve Bayes classifiers [34] were used in our experiments (see Section 2.2.1 for these algorithms). For SVM, a feature vector consisted of weighted frequencies of selected keywords, where we used the widely used TF-IDF weighting method [35] and then normalized vectors to unit vectors. The dot-product linear kernel was employed, and the default misclassification cost in LibSVM ($C = 1.0$) was used. For both classification algorithms, we trained multiple classifiers using different settings: different numbers of word features selected for IG, $n_f \in$

{100, 300, 900, …, 72,900} and different sizes of (likely-) negative instances randomly sampled among 42,302 unlabeled articles, $n_{neg} \in \{42{,}302 \times 1/2^0, 42{,}302 \times 1/2^1, 42{,}302 \times 1/2^2, \ldots, 42{,}302 \times 1/2^8 (\approx 165)\}$. IG is a commonly used filtering approach to feature selection for text classification [26,30–32,36] (see Section 2.2.2). In our preliminary experiment, we observed that SVM classifiers with features selected for IG generally performed as good as or better than SVM classifiers with the same number of features selected using other selection methods such as χ^2 statistics and point-wise mutual information. This result conformed to the previous studies on feature selection for text classification [30,31,36], and we chose to use IG in our study. We explored diverse parameters not only to seek good settings for the two classification algorithms, but also to derive a test corpus with diverse articles through the pooling method. Individual words that appeared in more than three articles were considered for features, while 571 stop words defined in the SMART project were pre-filtered.[4] For a particular choice of n_{neg} and n_f, not as many n_f features may be derived in the training corpus, and thus n_f is the upper bound of the word features. For example, for the smallest setting of n_{neg}, the number of available features was about 10,000, while for the largest setting of n_{neg}, it was about 60,000. To measure reliable performance for a classifier trained on sampled unlabeled instances, 10 sets of randomly sampled unlabeled articles were compiled (except for the case when all (likely-) negative instances were used, i.e., $n_{neg} = 42{,}302 \times 1/2^0$). Given a new article, it is assigned with a mean of 10 scores from 10 classifiers each trained using a different set of unlabeled articles. In other words, we developed an ensemble classifier consisting of 10 constituent classifiers using weighted voting [37]. To avoid calculating IG values of words for each setting of n_{neg}, the IG values were calculated once using all of the (likely-) negative instances, and the top n_f features that could be found in the under-sampled training corpus were used in each classifier. In summary, we trained naïve Bayes and SVM ensemble classifiers with nine different n_{neg} values[5] and seven different n_f values. Therefore, we trained $63 = 9 \times 7$ classifiers for naïve Bayes and SVM each, and a total of $126 = 63 \times 2$ classifiers were obtained.

4. Results and discussion

In evaluating machine learning classifiers, cross-validation tests are commonly used, in which a class-labeled data set is partitioned and classifiers are trained and evaluated on different partitions in a round-robin manner. The corpus initially compiled for training classifiers, however, was not suitable for cross-validation testing because articles were labeled by analysts during the course of their regular surveillance work and it may not be a representative article set in the application domain. Moreover, randomly sampled articles were regarded as negative instances in this corpus without being reviewed by analysts.[6] In addition, cross-validation testing involves shuffling of articles published over time when deriving a training and an evaluation corpus, which was not desirable when examining article classifiers for biosurveillance, e.g., articles to be classified by a classifier (articles in the evaluation corpus) should be only those published later than articles in the training corpus.

Using the broad Boolean query strings used to prepare the training corpus, 15 thousand articles published by South Asian sources were gathered over 15 days immediately following the one month period of the training data collection. As in the previous section, all

[4]The ftp://ftp.cs.cornell.edu/pub/smart/english.stop.

[5]For eight of the nine settings, ten classifiers were trained using differently sampled negative instances each time.

[6]The goal of the current study is to test if randomly sampled articles can be regarded as negative instances to train a reasonable text classifier in this domain. Such a corpus consisting of labeled relevant and unlabeled articles, however, is not suitable to test the performance of a text classifier because we cannot measure the performance precisely with unlabeled articles in it.

the trained classifiers were applied to this data set, and high-priority articles were identified for individual classifiers daily. We compiled top *N* high-priority articles for each day for a varying *N*, where we chose $N = 10$ so that the resulting set of articles were amenable to manual review by an analyst. Note that a classifier may assign the exact same score to two or more articles. For example, in our experiment, a naïve Bayes classifier often assigned 1.0 (the highest score) to more than 10 articles on a particular day. Therefore, for $N = 10$, we in fact compiled 10 or more than 10 articles for each classifier in case of ties. The resulting set of articles compiled over the 15 days contained 1337 articles from nearly 100 sources. These articles were reviewed by five Argus analysts including the one in charge of the target regions in South Asia, and the agreement measured using multi-κ statistics [38] was above 0.7, which is considered "substantial" agreement [39]. The disagreement could be attributed to an under-specification of target geographic regions in the article labeling guidelines prepared for this experiment. Therefore, we decided to use only the class labels assigned by the analyst in charge of the target regions. As a result, 564 (42%) were identified as relevant. The number of relevant articles (37.6 articles/day on average) was larger than what was anticipated. This finding was attributed to a natural disaster taking place in the region during the period, causing heightened concern for health-related issues. We also attribute this large number of relevant articles to the coverage of topics by Argus which are not direct reports of infectious disease outbreaks, such as enviro-climatic factors and indirect indicators of disease outbreaks. Fig. 3 shows the total number of articles retrieved daily and also the number of those labeled as relevant and irrelevant by the analyst.

4.1. Performance of diverse machine learning classifiers

For each of 15 days, AUCs of different classifiers were calculated using manually labeled articles. The highest mean-daily-AUC (average AUC over 15 days) among the SVM classifiers was 0.836 (observed for a classifier trained with 165 (likely-) negative instances and 24,300 features) and that among the naïve Bayes classifiers was 0.841 (observed for a classifier trained with 165 (likely-) negative instances and 2700 features). Fig. 3 shows the daily AUCs of these best performing naïve Bayes and SVM classifiers. Fig. 4 show the mean-daily-AUCs of all the naïve Bayes and SVM classifiers.

As in Fig. 4, for both naïve Bayes and SVM classifiers, a large set of (likely-) negative instances did not seem to improve individual classifiers. Unless an extremely large set was used, SVM classifiers performed better when all of the word features were used (Fig. 4B). This conforms to the known property of SVM that it can exploit a large number of features. As for naïve Bayes classifiers, classification performance was hardly affected by the size of negative instances when 2700 or less word features were used, and peak performance, competitive with the best SVM classifiers, was achieved when 900 to 2700 word features were used. More than 2700 word features, however, degraded the performance of naïve Bayes classifiers. Naïve Bayes classifiers, provided with an appropriate set of word features and any number of negative instances, perform well and are stable. This may imply that if articles contain certain keywords, a naïve Bayes classifier always assigns high scores to them. This could justify the use of extensive Boolean searches by Argus where keywords have been expanded and queries have been refined over time, but that in turn may undermine the practical utility of naïve Bayes classifiers in complementing manual Boolean searches.

Fig. 5 shows ROC curves for the SVM and the naïve Bayes classifiers that achieved the highest mean-daily-AUCs, where the ROC curves were drawn based on 1337 labeled articles. The naïve Bayes classifier assigned the score of 1.0 to 293 of 1337 articles (19.5 articles/day on average). This could be due to the property of naïve Bayes classifiers that

they tend to return output scores near 1.0 or 0.0 [24] as well as to the significant digits set for the outputs in Weka.

4.2. Sensitivity of classifiers in operational biosurveillance

In order to gain insight into the true sensitivity of automated classifiers in operational biosurveillance, we retrieved incident records from an Argus database on disease incidents. This database has been carefully maintained by expert analysts as part of the ongoing surveillance project independently from the current retrospective study. Among incidents reported during the test period, we randomly selected 10 incidents for our manual review. Specifically, we checked if these incidents could be found among the high-priority articles identified by any of the trained classifiers. Table 1 shows the descriptions of the selected incidents and the number of 126 trained classifiers that identified an article(s) reporting each of these incidents as daily top 10 high-priority articles. The numbers of classifiers in Table 1 may imply which incidents were easy/difficult to detect, but the criteria for an article to qualify for the top 10 depends on many other factors, e.g., what other important incidents were taking place on the same day. Table 2 shows the ranks of the articles reporting these incidents according to the best performing naïve Bayes and SVM classifiers that yielded the highest mean-daily AUCs. Notably all of the 10 incidents were found as daily top 10 high-priority articles by at least two of the trained classifiers, e.g., if we had formed an ensemble of 126 classifiers and combined their daily top 10 articles just as we did, we could capture all of these 10 incidents, while filtering 15 thousand articles down to 1337 (~9% of 15 thousand). An article reporting one of these incidents (I-7) was ranked within the daily top 10 by only two SVM classifiers that actually were trained using all of the unlabeled instances available. No naïve Bayes classifier listed this particular article within the daily top 10. We may hypothesize that diverse SVM classifiers trained using large (likely-) negative instances can be useful in complementing analysts' article searches despite their limited overall performance. Further investigation is needed to test this hypothesis in the future.

5. Related work

There have been several studies addressing text classification for Internet-based biosurveillance as described later [3,8,11–13]. Zhang and Liu [13] explored detection of sentences containing disease outbreak information in ProMED-mail.[7] They addressed the issue that the vocabulary used to report disease outbreaks overlaps with that used in other public health news, such as the treatment of diseases, and therefore text classifiers based on standard word features would not be effective in identifying disease outbreak reports. They used a dependency parser to identify sentence structures, and extracted verbs and adjectives directly modifying disease names as features. Tense and negation status of verbs were also detected. In addition, word n-grams (consecutive n words in text) and time expressions (e.g., "today", "three months ago", etc.) were extracted from sentences as additional features. Naïve Bayes and SVM classifiers using these features were evaluated on 2342 ProMED-mail sentences (1660 positive and 682 negative) in cross-validation tests. The two classification algorithms were found to be competitive on this data set, and the performance of the both types of classifiers could be improved with the proposed features (*F*-score[8] improved, roughly, from 0.7 to 0.76).

The BioCaster project, aimed at providing automated detection and tracking of disease outbreak information based on online news articles, has studied text classification for

[7] A mailing list on infectious disease and toxins hosted by the International Society for Infectious Diseases.

[8] *F*-score is a harmonic mean of precision and recall (sensitivity), where precision is the ratio of true positives to all predicted positives (true positives + false positives), and recall is the ratio of true positives to all positives (true positives + false negatives).

automated filtering of relevant articles. Doan et al. [11] investigated the utility of named entity features (e.g., person, location, organization, disease, time, and 13 other types of named entities) supplemented with "role" and "quality" information (e.g., case and number of a noun phrase specifying role/quality of the referent in the given context). Naïve Bayes classifiers with and without using these features were evaluated on 500 news articles (231 positive and 269 negative) in cross-validation tests. Improved performance was reported for classifiers using these additional features (the precision was improved from 0.64 to 0.75, where the recall was 1.0). Conway et al. extended this work to report the utility of semantic labels assigned to words as well as word *n*-grams that further improved the classification performance, where they also reported the importance of feature selection [12].

These studies primarily focused on the use of additional feature types beyond bag-of-words for improving classification performance. Meanwhile, the focus of our current study is to examine a non-intrusive biosurveillance framework, and our current results are not comparable to the results reported in these existing studies. In the future, we plan to investigate these rich feature types in the proposed framework.

A variant of Bayesian filters and 'a proprietary algorithm' were used, respectively, in the HealthMap project [3] and in the GPHIN project [8]. To our knowledge, however, information of the training corpora or classification performance has not been reported for these systems.

6. Conclusion

In this study, we tested a non-intrusive approach to training automated text classifiers for Internet-based biosurveillance. Naïve Bayes and SVM classifiers were trained using different sizes of unlabeled instances and word features. The performance of classifiers trained in the proposed framework was promising, and will likely improve over time as more labeled articles are accumulated in this framework. The best performance observed for the two algorithms, naïve Bayes and SVM, was competitive, while the classifiers trained with varying settings behave differently for the two algorithms. Naïve Bayes classifiers, provided with an appropriate number of word features, performed well for almost any number of unlabeled instances. We believe this stable performance of naïve Bayes classifiers is due to the presence of a keyword set strongly implicative of positive (i.e., relevant) articles and hypothesize that naïve Bayes classifiers are not suitable for identifying articles reporting subtle indications with no explicit keywords. Unlike naïve Bayes classifiers, when the settings were varied, SVM classifiers ranked articles differently and, despite the poor performance of some, they together seemed to yield a robust ensemble classifier covering diverse positive articles.

Automated text classification is an important technology in Internet-based biosurveillance projects [11–14,21]. In Project Argus, machine learning text classifiers have been used to prioritize a large volume of articles retrieved by surveillance analysts with Boolean queries. Development and maintenance of those text classifiers require analysts to label a sufficient number of "irrelevant" articles besides "relevant" articles, which is an expensive process in the time-sensitive operational surveillance workflow. The results of our current study suggest that a classifier can be built and maintained without requiring explicit labeling of irrelevant articles. In particular, the retrospective study described in Section 4.2 shows that incidents detected in a real-life biosurveillance project could be detected effectively using the proposed approach. Considering the time and cost, we believe the proposed framework has significant advantages over traditional approaches in building text classifiers for a constantly evolving domain (Fig. 1). We plan to implement the approach and further evaluate its effectiveness in the real-time surveillance setting.

Additionally, several intriguing research topics become apparent when introducing the proposed framework to Internet-based biosurveillance operations. One research topic is the impact of recent articles in maintaining text classifiers. We believe that the distribution of online articles constantly changes and classifiers in this domain should be updated frequently using an up-to-date corpus. A large training corpus could be available with the proposed framework and the effective use of such a corpus should be investigated in the future. Another research topic is the impact of the size of the available training corpus (positive articles). A preferred classification algorithm can depend on the size of an available training corpus [40]. The current results on naïve Bayes and SVM classifiers may not hold as more labeled positive articles become available. Finally, the most challenging topic is the detection of rare incidents. A classifier yielding good overall performance must focus on the detection of frequent types of incidents, while it may overlook articles reporting rare types. We plan to investigate these topics over a long time period in our future study.

Acknowledgments

This research and development project was conducted by Georgetown University and is made possible by a contract awarded and administered by the U.S. Army Medical Research and Materiel Command (USAMRMC) and the Telemedicine and Advanced Technology Research Center (TATRC), Fort Detrick, Maryland 21702, under contract number W81XWH-04-1-0857. The views, opinions and findings contained in this research are those of the authors and do not necessarily reflect the views of the Department of Defense and should not be construed as an official DoD/Army policy unless so designated by other documentation. No official endorsement should be made. We thank our sponsors and also the members of Project Argus, especially Mr. Dan Ji and Mr. Peter Li for their technical support and Dr. Kevin Jones for stimulating discussion.

References

1. Keller M, Blench M, Tolentino H, Freifeld CC, Mandl KD, Mawudeku A, et al. Use of unstructured event-based reports for global infectious disease surveillance. Emerg Infect Dis. 2009
2. Collier N, Doan S, Kawazoe A, Goodwin RM, Conway M, Tateno Y, et al. BioCaster: detecting public health rumors with a Web-based text mining system. Bioinformatics. 2008; 24:2940–2941. [PubMed: 18922806]
3. Brownstein JS, Freifeld CC, Reis BY, Mandl KD. Surveillance Sans Frontiers: Internet-based emerging infectious disease intelligence and the HealthMap project. PLoS Med. 2008; 5:e151. [PubMed: 18613747]
4. JRC. Health Emergency & Disease Information System: a web based platform for the Public Health Community. 2007. http://langtech.jrc.it/Documents/0711FlyerHEDIS-Medisys-web.pdf
5. Amato-Gauci A, Ammon A. The surveillance of communicable diseases in the European Union – a long-term strategy (2008–2013). Euro Surveill. 2008; 13
6. Damianos L, Ponte J, Wohlever S, Reeder F, Day D, Wilson G, et al. MiTAP, text and audio processing for bio-security: a case study. Eighteenth National Conference on Artificial Intelligence. 2002:807–814.
7. Hartley DM, Nelson NP, Walters R, Arthur R, Yangarber R, Madoff L, et al. The landscape of international event-based biosurveillance. Emerg Health Threats J. 2010; 3
8. Blench, M. Global Public Health Intelligence Network (GPHIN). Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas (AMTA); Waikiki, Hawaii. 2008. p. 299-303.
9. WHO. Severe acute respiratory syndrome (SARS): status of the outbreak and lessons for the immediate future. 2003. http://www.who.int/csr/media/sarswha.pdf
10. Wilson JM, Polyak MG, Blake JW, Collmann J. A heuristic indication and warning staging model for detection and assessment of biological events. J Am Med Inform Assoc. 2008; 15:158–171. [PubMed: 18096906]
11. Doan, S.; Kawazoe, A.; Collier, N. BioNLP 2007. Prague; Czech Republic: 2007. The role of roles in classifying annotated biomedical texts; p. 17-24.

12. Conway, M.; Doan, S.; Kawazoe, A.; Collier, N. Classifying disease outbreak reports using n-grams and semantic features. The Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008); Finland. 2008. p. 29-36.
13. Zhang, Y.; Liu, B. Knowledge Discovery in Databases: PKDD 2007. Springer; Berlin/Heidelberg: 2007. Semantic text classification of emergent disease reports; p. 629-637.
14. Zhang YL, Dang Y, Chen HC, Thurmond M, Larson C. Automatic online news monitoring and classification for syndromic surveillance. Decis Support Syst. 2009; 47:508–517.
15. Walters, R.; Harlan, P.; Nelson, NP.; Hartley, DM. Data sources for biosurveillance. In: Voeller, JG., editor. Wiley Handbook of Science and Technology for Homeland Security: Risk Analysis. 2009.
16. Nelson NP, Brownstein JS, Hartley DM. Event-based biosurveillance of respiratory disease in Mexico, 2007–2009: connection to the 2009 influenza A(H1N1) pandemic? Euro Surveill. 2010; 15 Available online: http://www.eurosurveillance.org/ViewArticle.aspx?ArticleId=19626(pii=19626).
17. Graham, P. A plan for spam. 2002. http://www.paulgraham.com/spam.html
18. Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, et al. PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinform. 2003; 4:11.
19. Noto K, Saier MH, Elkan C. Learning to find relevant biological articles without negative training examples. Ai 2008: Advances in Artificial Intelligence, Proceedings. 2008; 5360:202–213.
20. Chapman WW, Dowling JN, Wagner MM. Fever detection from free-text clinical records for biosurveillance. J Biomed Inform. 2004; 37:120–127. [PubMed: 15120658]
21. Freifeld CC, Mandl KD, Reis BY, Brownstein JS. HealthMap: global infectious disease monitoring through automated classification and visualization of Internet media reports. J Am Med Inform Assoc. 2008; 15:150–157. [PubMed: 18096908]
22. Hand DJ, Yu K. Idiot's Bayes – not so stupid after all? Int Stat Rev. 2001; 69:385–399.
23. McCallum, A.; Nigam, K. A comparison of event models for Naive Bayes text classification. AAAI-98 Workshop on Learning for Text Categorization; 1998. p. 41-48.
24. Bennet, PN. Assessing the Calibration of Naive Bayes' Posterior Estimates. School of Computer Science, Carnegie Mellon University; Pittsburgh, PA: 2000.
25. Vapnik, V. Statistical Learning Theory. Wiley-Interscience; 1998.
26. Joachims, T. Making large-Scale SVM Learning Practical. Advances in Kernel Methods – Support Vector Learning. MIT-Press; 1999.
27. Yang Y, Liu X. A re-examination of text categorization methods. ACM Special Interest Group of Information Retrieval (SIGIR). 1999:42–49.
28. Chang, C-C.; Lin, C-J. LIBSVM: a library for support vector machines. 2009. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/
29. Scholkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural Comput. 2001; 13:1443–1471. [PubMed: 11440593]
30. Yang, Y.; Pedersen, JO. A comparative study on feature selection in text categorization. Fourteenth International Conference on Machine Learning; 1997. p. 412-420.
31. Forman G. An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res. 2003; 3:1289–1305.
32. Manning, CD.; Raghavan, P.; Schütze, H. Introduction to Information Retrieval. Cambridge University Press; Cambridge, New York: 2008.
33. Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett. 2006; 27:861–874.
34. Witten, IH.; Frank, E. Data Mining: Practical Machine Learning Tools and Techniques. 2. Morgan Kaufmann; San Francisco: 2005.
35. Sparck-Jones K. A statistical interpretation of term specificity and its application in retrieval. J Doc. 2004; 60:493–502.
36. Yin L, Xu G, Torii M, Niu Z, Wu C, Hu Z, et al. Document classification for mining host pathogen protein–protein interactions. Artif Intell Med. 2010; 49(3):155–160. [PubMed: 20472411]
37. Dietterich TG. Ensemble methods in machine learning. Mult Classifier Syst. 2000; 1857:1–15.

38. Artstein R, Poesio M. Inter-coder agreement for computational linguistics. Comput Linguist. 2008; 34:555–596.

39. Landis JR, Koch GG. The measurement of observer agreement for categorical data. Biometrics. 1977; 33:159–174. [PubMed: 843571]

40. Ng, AY.; Jordan, MI. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. Neural Information Processing Systems Conference (NIPS); 2002. p. 841-848.

Summary points

What is known about the subject

- Online news articles can provide timely information on disease outbreaks worldwide, though it is not trivial to identify these outbreaks among a large volume of online publications.
- Automated text classification using machine learning is a promising approach to identifying online news articles relevant to disease outbreaks, where a machine learning system needs to be trained on a set of manually labeled relevant and irrelevant articles.

What this study adds

- A practical text classification system can be trained on manually labeled relevant articles and unlabeled articles without requiring manually labeled irrelevant articles in an Internet-based biosurveillance project.
- A training data set for text classification systems (labeled relevant articles) can be compiled in a real-time operation of Internet-based biosurveillance without disturbing analysts' regular workflow.
- In an operational Internet-based biosurveillance project, a large training set for a machine learning text classification system can be compiled over time, and a classification system may be retrained on an up-to-date training data set periodically.

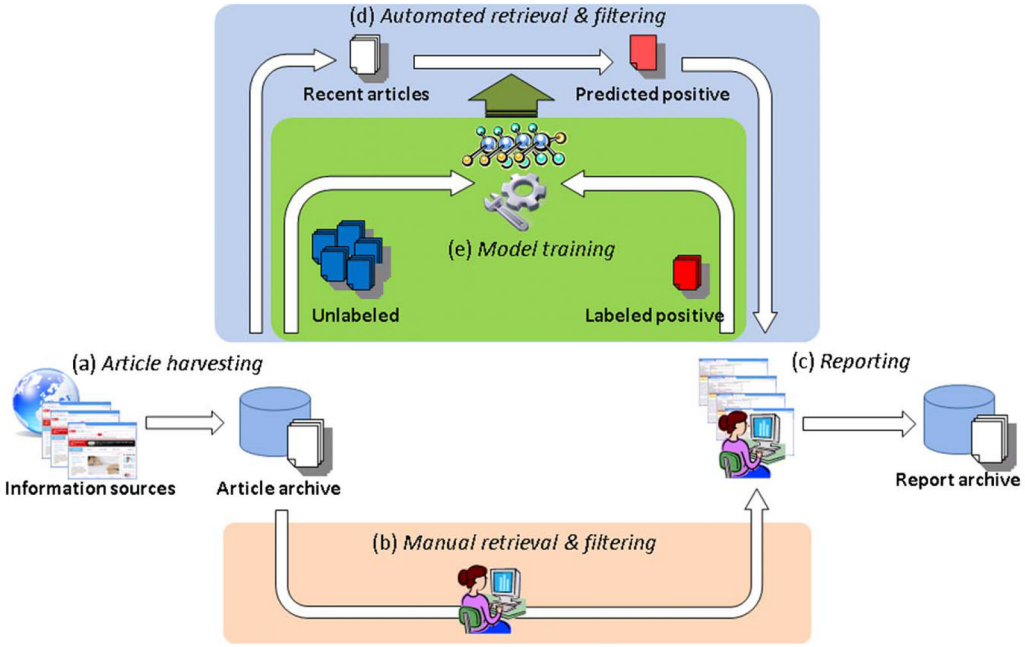

Fig. 1.

Overview of the Internet-based biosurveillance workflow. Automated retrieval and filtering of online articles, (d), is proposed to complement the manual effort, (b). An ensemble system to facilitate automated text classification can be trained periodically on labeled positive and unlabeled articles gathered in a non-intrusive manner, (e).

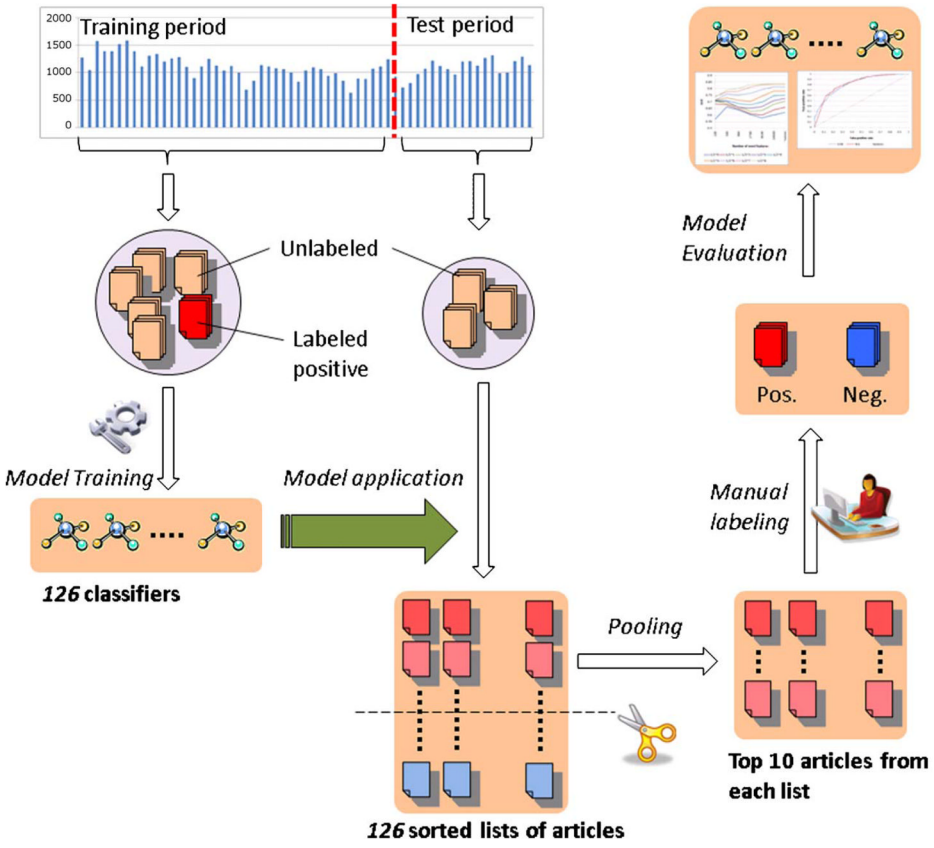

Fig. 2.

Overview of the evaluation framework. Classifiers trained on articles published during the training period were applied to articles published during the test period. High-priority articles identified by each classifier were *pooled* and then classified by an Argus analyst. Only the articles reviewed by the analyst were used for the evaluation of the classifiers.

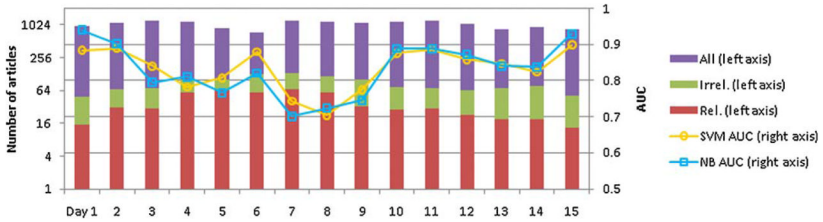

Fig. 3.

The numbers of retrieved articles and the performance of classifiers. The number of all articles retrieved for a broad Boolean search (All) and the numbers of articles manually labeled as relevant (Rel.) and irrelevant (Irrel.). Daily AUCs of the best performing SVM and naïve Bayes classifier are also shown (see Section 4.1).

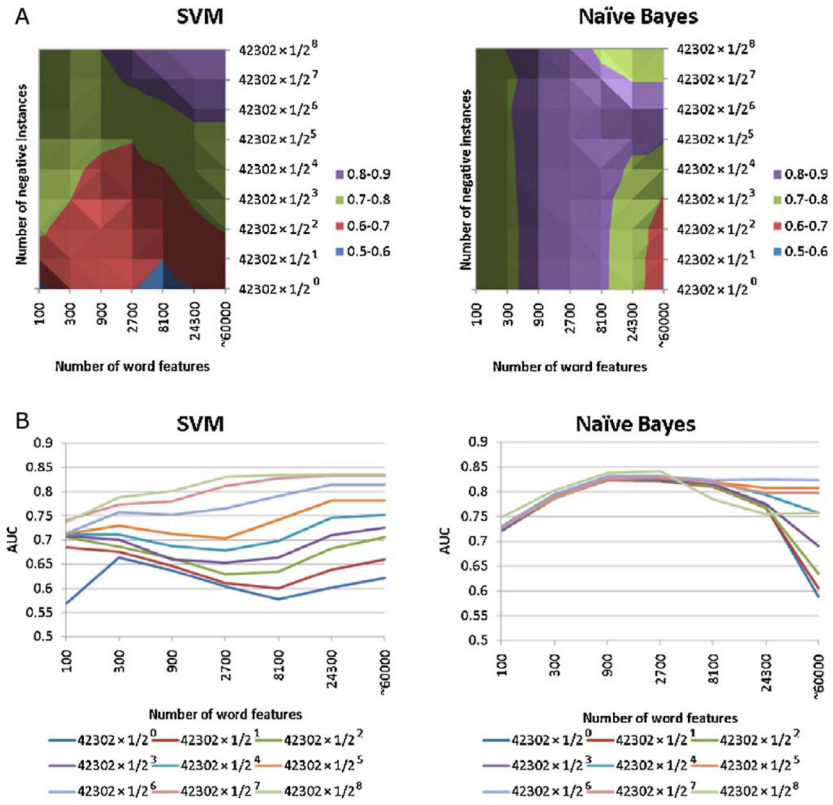

Fig. 4.

Daily averages of AUCs for SVM and naïve Bayes classifiers. The same information is presented in two different formats. (A) Contours of AUC observed for classifiers trained with different pairs of the parameters, which are the number of negative instances in the training corpus and the number of word features selected based on information gain (IG). (B) Line graphs showing changes in AUC at different numbers of word features selected for IG, where different lines/colors are used for different numbers of negative instances in the training corpus.

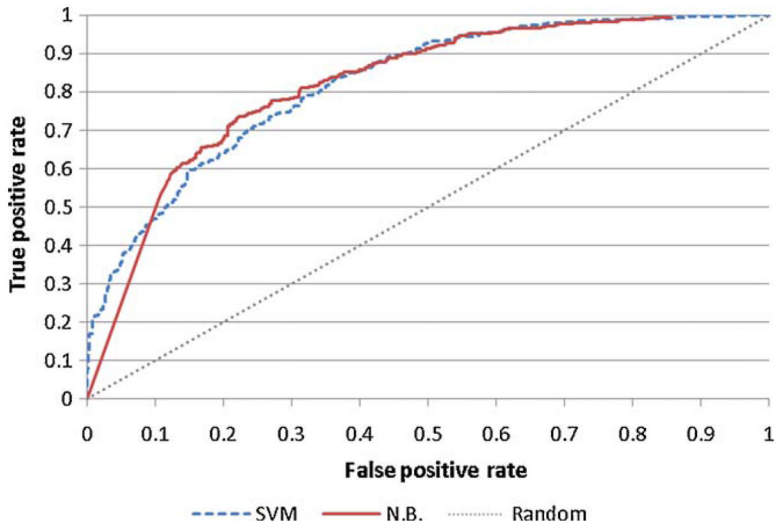

Fig. 5.

ROC curves of the SVM classifier and naïve Bayes classifier. The SVM classifier is trained using $1/2^8 \times 42{,}302$ negative instances and 24,300 features and naïve Bayes (N.B.) classifier is trained using $1/2^8 \times 42{,}302$ negative instances and 2700 features. The curves are based on 1337 articles manually reviewed and labeled by an analyst. The straight line at the small false positive rates for the N.B. classifier is due to multiple articles all assigned with output scores of 1.0.

Table 1

Ten incidents reported in Argus during the evaluation period. This table shows how many of 126 classifiers (63 SVM and 63 naïve Bayes classifiers) identified articles reporting the incidents as daily top 10 high-priority articles.

ID	Day	Description of incidents	Classifier counts	
			SVM	NB
I-1	1	People were hospitalized for an unknown cause.	44	43
I-2	2	People died from a disease.	55	61
I-3	3	People were diagnosed with a disease.	56	50
I-4	4	A natural disaster took place.	5	41
I-5	6	A hospital reported a situation about a disease.	7	45
I-6		People were diagnosed with a disease.	24	56
I-7	7	A local government banned imports of a product.	2	0
I-8	8	People died from a disease.	40	54
I-9		People were diagnosed with a disease.	28	58
I-10	14	People died after a vaccination.	12	60

Table 2

Ten incidents reported in Argus during the evaluation period. This table shows the daily rank of the articles reporting each event. The numbers in the parentheses are the number of tied articles for the rank, e.g., “1 (15)” indicates that an article reporting a particular incident was ranked first, while there were 14 other articles that were assigned with the same score (these 15 articles may or may not report the same incident). The last column (Max) shows the number of all the articles retrieved on a specific day.

ID	Day	Description of incidents	Rank (# of ties)		Max
			SVM	NB	
I-1	1	People were hospitalized for an unknown cause.	4	1 (15)	957
I-2	2	People died from a disease.	3	1 (21)	1089
I-3	3	People were diagnosed with a disease.	3	1 (24)	1208
I-4	4	A natural disaster took place.	34	1 (39)	1168
I-5	6	A hospital reported a situation about a disease.	40	1 (48)	736
I-6		People were diagnosed with a disease.	26	1 (48)	
I-7	7	A local government banned imports of a product.	191	1 (183)	1193
I-8	8	People died from a disease.	16	1 (50)	1187
I-9		People were diagnosed with a disease.	5	1 (50)	
I-10	14	People died after a vaccination.	23	1 (18)	941