Detecting Healthcare-Associated Infections in Electronic Health Records - Evaluation of Machine Learning and Preprocessing Techniques

Claudia Ehrentraut[1], **Maria Kvist**[1,2], **Elda Sparrelid**[3] **and Hercules Dalianis**[1]
[1]Department of Computer and Systems Sciences (DSV), Stockholm University, Sweden
[2]Department of Learning, Informatics, Management and Ethics (LIME), Karolinska Institutet, Sweden
[3]Department of Infectious Diseases, Karolinska University Hospital, Sweden
ehrentraut@dsv.su.se, maria.kvist@karolinska.se
elda.sparrelid@karolinska.se, hercules@dsv.su.se

Abstract

Healthcare-associated infections (HAI) are infections that patients acquire in the course of medical treatment. Being a severe public health problem, detecting and monitoring HAI in healthcare documentation is an important topic to address. Research on automated systems has increased over the past years, but performance is yet to be enhanced. The dataset in this study consists of 214 records obtained from a Point-Prevalence Survey. The records are manually classified into HAI and NoHAI records. Nine different preprocessing steps are carried out on the data. Two learning algorithms, Random Forest (RF) and Support Vector Machines (SVM), are applied to the data. The aim is to determine which of the two algorithms is more applicable to the task and if preprocessing methods will affect the performance. RF obtains the best performance results, yielding an F_1-score of 85% and AUC of 0.85 when lemmatisation is used as a preprocessing technique. Irrespective of which preprocessing method is used, RF yields higher recall values than SVM, with a statistically significant difference for all but one preprocessing method. Regarding each classifier separately, the choice of preprocessing method led to no statistically significant improvement in performance results.

1 Introduction

1.1 Healthcare-associated infections

The Centers for Disease Control and Prevention (CDC) define healthcare-associated infections (HAI) as "infections that patients acquire during the course of receiving healthcare treatment for other conditions." (CDC, 2013)

HAIs pose a public health problem worldwide, in developed as well as resource-poor countries. A survey conducted under the patronage of the World Health Organisation (WHO), in 55 hospitals from 14 countries, found that about 8.7% of all hospital inpatients suffered from HAI. Besides being a leading cause of death among hospitalised patients, HAIs impose an enormous economic burden on healthcare facilities due to, for instance, a prolonged stay of the infected patients or increased drug usage (Ducel et al., 2002).

1.2 Automatic surveillance of healthcare-associated infections

Detecting and monitoring HAI is an important topic for healthcare to address. Many different attempts to confine HAIs have been made, e.g., better hygiene (Breathnach, 2009) and manual surveillance performed by infection control professionals (Evans et al., 2009). Nevertheless, the presence of HAIs persists in modern health facilities.

The increasing amount of digital data available in hospitals has pioneered the way for and led to an increase in research and the development of automated systems over the past years (Adlassnig et al., 2009). MONI-ICU (Blacky et al., 2011) and the HELP system (Evans et al., 2009) are examples of automated systems, which are designed to detect HAI, that are in use. For most systems, however, performance still needs to be enhanced before these systems can be effectively established in real-life hospital settings.

Automated surveillance systems are either based on handcrafted rules describing medical knowledge, or on machine learning algorithms. Machine learning describes the task of finding patterns that underlie some example data, and using them in order to make predictions on unseen data.

1.3 Aim

The study focuses on applying machine learning techniques to the problem of detecting HAI. Two well-known algorithms, Random Forest (RF) and Support Vector Machines (SVM), are applied to the data. The aim is to determine which of the two algorithms is more applicable to the task, i.e., which of them performs best when detecting HAI.

In combination with each of the learning algorithms, we apply nine different preprocessing methods, including lemmatisation, reduction to infection-specific terms, and negation detection. Assuming that preprocessing will influence which features are selected by the classifier, we seek to answer the questions if and to what extent the preprocessing methods affect the performance of RF and SVM.

Another goal of this study is to investigate whether the presence of community-associated infections will affect the performance.

2 Data

2.1 Characteristics

The dataset[1] encompasses electronic health records (EHR) from 120 inpatients at a major university hospital in Sweden and was collected during a Point-Prevalence Survey (PPS)[2] in spring 2012.

Not all information stored in the patients' EHRs was considered valuable by the physicians for detecting HAI. Thus, a subset of information from the EHRs was retrieved: Journalanteckning (Engl.: record notes), Läkemedelsmodul (Engl.: drug module), Mikrobiologiska Svar (Engl.: microbiological result), and Kroppstemperatur (Engl.: body temperature). The information extracted from these modules consists of structured and unstructured data. Structured data refers to data that is stored in predefined fields, such as ICD-10 diagnosis codes, medication or body temperature. Unstructured data refers to textual notes written by physicians, such as daily notes or microbiological results.

For each of the 120 patients, information from all four modules was extracted for the patient's entire hospitalisation. The physicians define one hospitalisation as the stay of a patient at a health facility for one care process. If the patient is discharged from one department of the hospital and admitted to another within 24 hours, this is regarded as the same hospitalisation. Moreover, any noted event occurring within 24 hours after discharge is included in the hospitalisation. From this point on, we will refer to the file that contains the data of a patient's entire hospitalisation as the Hospitalisation Record (HR). Since some of the 120 patients were hospitalised multiple times during the five month-period of records we received, our dataset comprises 214 HRs. Hospitalisations of less than 48 hours are not represented in the final dataset as they were considered to carry too little information. Table 1 depicts the characteristics of the HRs that are used as input for the classifiers.

Table 1: Characteristics of the patient records used.

	HAI	NoHAI	Total
Number of HRs	131	83	214
Length in days	2-144	3-93	2-144
Number of words	1,536,656	311,729	1,848,385

2.2 Class distribution

120 patients had experienced HAI according to the PPS results. We had access to a five months period of records. As a result, the physicians in this study, unlike the physicians who carried out the PPSs, obtained information on how the health status of the patient progressed and which assessment he or she received during the time after the PPSs had been conducted. The physicians in this study could therefore give a more accurate answer on whether HAI occurred or not. Only 131 of 214 HRs contained HAI diagnoses (positive examples). The remaining 83 HRs contained no HAI diagnoses (negative examples), according to their assessment. The dataset is not balanced but instead skewed towards the pos-

[1] This research has been approved by the Regional Ethical Review Board in Stockholm (Etikprövningsnämnden i Stockholm), permission number 2012/1838-31/3.

[2] In Sweden, PPSs are performed twice a year to estimate the occurrence of HAI by counting existing cases of HAI at one specific time.

itive class, with a skew value[3] of 0.634. Besides being unbalanced, the classes HAI and NoHAI contain HRs of mixed kind. Some of those that contain HAI also contain community-associated infections (CAI). As depicted in Table 2, 11 of the HRs containing HAI also contain an additional CAI. The class NoHAI comprises HRs that do not contain HAI. Of these, 25 contain CAI while 58 HRs do not include any infections at all (NoINF).

Table 2: Distribution of CAI and NoINF among HAI and NoHAI class.

Class	HAI	CAI	NoINF	Total
HAI	131	11	-	131
NoHAI	-	25	58	83

3 Method

3.1 Automatic text classification

Text classification describes the task of classifying documents into predefined classes by means of a machine learning algorithm (Hoyt et al., 2012). In the learning phase, the algorithm is trained on a training set, where each document is labelled with the class it belongs to. Based on this training data, a classification model is built by identifying common core characteristics of all documents in each class. The built model is then used to assign unseen documents to one of the predefined classes. In this study, we deploy the machine learning algorithms RF and SVM to classify the HRs into the predefined classes HAI and NoHAI. Both algorithms are used within the WEKA[4] environment.

3.1.1 Random Forest (RF)

The core aspect of RF is that it constructs multiple single decision trees, i.e., weak learners, to form one stronger learner (Breiman, 2001). The challenges in our data lie in the fact that the sample is small and imbalanced while the feature space is high-dimensional, each feature carrying only a small amount of information. RF has become a popular choice within bioinformatics, precisely because the algorithm is able to deal with these aspects (Qi, 2012). The following parameter settings are chosen for RF. The number of trees is set to 1000. The number of features that are randomly chosen at each node of the tree is 10.

3.1.2 Support Vector Machines (SVM)

SVM uses the concept of representing documents that are to be classified as points in a high-dimensional space, and finding the line that separates them. SVM tries to find the line with the maximum margin, where margin refers to the distance between the line and the nearest data points (Noble, 2006). Using SVM is, among other reasons, motivated by the statement that SVM is found to be very effective for 2-class classification problems (Dalal and Zaveri, 2011). We use a non-optimised SVM with a radial basis function (RBF) kernel with degree=3, C=1, epsilon=0.001 and gamma=1/1,000.

3.2 Text representation

Like many other classifiers in WEKA, RF and SVM cannot handle data that is represented as a string. Therefore, the HRs need to be converted into a format that is readable for the classifiers. The HRs in the dataset are represented as bag-of-words vectors when being passed as input to the classifiers. In this representation, which is the most prevalent one for text classification (Boulis and Ostendorf, 2005), the order of words is ignored (Manning et al., 2008). Consequently, syntactical information as well as the linear temporality of events are not preserved.

3.3 Nine preprocessing methods

The data was preprocessed in nine different ways: *NoStopwords*, *Lemma*, *Infection-Specific Terms (IST)*, *NegationTagged*, *NegationTagged + IST*, *NegationRemoved*, *NegationRemoved + IST*, *TF-IDF 50*, and *Tagged*. After preprocessing, the 1,000 most frequent terms were selected from each version of the data, and their corresponding TF-IDF[5] values were stored in the frequency vectors. A non-preprocessed version of the dataset was kept as a baseline and termed *Plain*. For the method *TF-IDF*

[3] The skew is measured according to the definition given in (Jeni et al., 2013).

[4] WEKA is a collection of state-of-the-art machine learning algorithms, data preprocessing and visualisation tools. `http://www.cs.waikato.ac.nz/ml/weka/`.

[5] TF-IDF stands for Term Frequency-Inverse Document Frequency. It shows the importance of a feature in the text and is considered the most common method of weighting in text mining (Shi et al., 2011).

50, the 50 most frequent terms were kept from a non-preprocessed version of the dataset.

3.3.1 Stop word filtering

Stop words are terms that are regarded as not conveying any significant semantics to the texts they appear in and are consequently discarded (Dragut et al., 2009). Removing them from the text makes it easier for text processing systems to index the remaining text (Fox, 1989). In Swedish texts, stop words comprise 43% of the words (Sigurd, 1991). A Swedish stop word list that comprises 113 words, such as *och* (Engl.: and) and *att* (Engl.: to), was used.[6] The method where stop words are filtered out is called *NoStopwords*.

3.3.2 Lemmatisation

Lemmatisation is a method to reduce inflected words to their lemma or base form. For example, the inflected words *blödning, blödningar* (Eng.: bleeding, bleedings) are reduced to *blöda* (Eng.: bleed). We used the CST lemmatiser (Jongejan and Haltrup, 2013), which is adapted to Swedish. This method is called *Lemma*.

3.3.3 Reduction to infection-specific terms

The idea behind this attempt is to influence the feature selection process by reducing the HRs to infection-specific terms (ISTs), that is, to delete all other terms from the HRs. We consider ISTs to bear a significant part of the information, which is important to identify whether the HRs contain HAI or not. All ISTs are contained in a terminology, which was built in a semiautomatic approach. The physicians supplied a seed set of 27 ISTs. The seed set was extended by finding related terms, e.g., synonyms or misspellings of the input term, through the use of an automatic synonym generator based on Random Indexing[7] (Sahlgren, 2005). One physician then analysed the proposed terms with respect to whether they could be regarded as applicable ISTs or not. All relevant terms were added to the terminology. In the present approach, the terminology contained 2,201 terms. The terms in the terminology can be assigned to 6 main classes, which were created by the physicians: ATC[8] – 1108 terms, e.g., *tetracyklin* (Eng.: tetracycline), Action – 157 terms, e.g., *ultraljud* (Eng.: ultrasound), Diagnosis – 281 terms, e.g., *lunginflammation* (Eng.: pneumonia), Device – 101 terms, e.g., *kateter* (Eng.: catheter), Event – 340 terms, e.g., *intubering* (Eng.: intubation), Symptom – 214 terms, e.g., *feber* (Eng.: fever). The method where all but the infection-specific terms are removed from the HRs is referred to as *IST*.

3.3.4 Negation detection using NegEx

Clinical text contains many negations, e.g., negated symptoms or diagnoses. Negation detection is the technique to identify such negated entities. NegEx is a negation detection system for clinical text that has been adapted to Swedish (Skeppstedt, 2011). NegEx was run on our dataset to detect ISTs. 3,233 negated ISTs were found, which accounts for approximately 5% of the total of 60,932 ISTs in the data. The method where a tag was added to the negated term, e.g., *<NEGATED>infection<NEGATED>*, is called *NegationTagged*; where the negated term is removed, it is referred to as *NegationRemoved*. Both methods were also used in combination with the *IST* method, being termed *NegationTagged + IST* and *NegationRemoved + IST*, respectively.

3.3.5 Tagging

Tagging is performed automatically by using the ISTs and meta information, which could be easily identified automatically. ISTs are tagged with their respective class. For instance, the term *operation* belongs to the class event and is thus tagged as *<event>operation<event>*. Temperature values could be easily identified in the data and are tagged in the following manner: *<temp>38<temp>*. In addition, meta data, i.e., age, gender and length of hospital stay, is identified for each HR and annotated with the respective tag. The patient records are reduced to tagged terms only. This method is referred to as *Tagged*.

[6] `http://snowball.tartarus.org/algorithms/swedish/stop`.

[7] The Random Indexing model is trained on the first five months of patient records from 2008, which form a subset of the Stockholm EPR corpus (Dalianis et al., 2009).

[8] Anatomical Therapeutic Chemical Classification (ATC).

Table 3: Classification results (in percent) of the RF and SVM classifier in combination with the respective preprocessing methods. The best result of each classifier is highlighted.

	RF				SVM			
Preprocessing	Precision	Recall	F_1-score	AUC	Precision	Recall	F_1-score	AUC
Plain	80	86	83	0.84	80	71	75	0.71
NoStopwords	79	87	83	0.84	79	70	74	0.70
Lemma	**83**	**87**	**85**	**0.85**	79	71	75	0.70
IST	80	86	83	0.86	84	74	79	0.76
NegationTagged	80	89	84	0.84	78	71	74	0.70
NegationTagged + IST	81	86	84	0.86	83	73	79	0.75
NegationRemoved	80	87	83	0.84	79	70	73	0.69
NegationRemoved + IST	80	85	82	0.87	**86**	**74**	**80**	**0.77**
TF-IDF 50	74	79	77	0.80	72	65	68	0.63
Tagged	79	84	82	0.85	82	69	75	0.72

4 Evaluation

4.1 Measures

Although dealing with a binary classification problem, we are mainly interested in the HAI class. Performance of the classifier for this class is measured in recall, precision and F_1-score, which are standard measures when evaluating the performance of information systems (Van Rijsbergen, 1979). The measures can be defined in terms of true positives (TPs), false positives (FPs), true negatives (TNs) and false negatives (FNs). The area under the ROC curve (AUC) is considered during evaluation, since it provides an important value on whether there is a statistical dependence between the features and the class. An area of 1 represents a perfect test in separating two classes, while an area of 0.5 represents a worthless test (Tape, 2006). Our dataset is slightly imbalanced. Metrics such as Cohen's Kappa or accuracy are attenuated by skewed distributions, whereas AUC is not (Jeni et al., 2013).

4.2 10-fold cross-validation

For evaluation, stratified 10-fold cross-validation was used, one of the best known and most commonly used evaluation techniques (Japkowicz and Shah, 2011). One run of 10-fold cross-validation was performed.

4.3 Statistical tests

The non-parametric Sign Test (two-tailed, at 5% significance level) was used to test the significance of the results (Japkowicz and Shah, 2011).

4.4 Error analysis

For the method yielding the best result, the FPs were analysed with regard to how many of them contained CAI and NoINF respectively. Manual error analysis was performed by looking at the resulting TP, FP, TN and FN groups of HRs. The records were examined by one physician to determine whether there were any similarities from a clinical perspective.

5 Results

5.1 Classification results

Table 3 depicts the classifier performance for RF and SVM.

With regard to recall, RF outperforms SVM, irrespective of which preprocessing method is used. Except for the *TF-IDF 50* preprocessing technique, the recall values obtained by RF are statistically significantly higher than the ones obtained by SVM. On average, the recall obtained by RF is 86%, about 15 percentage points higher than for SVM.

With regard to precision, however, the difference in the performance of the two classifiers is not statistically significant, irrespective of which preprocessing method is used. Both classifiers have an average precision of 80% and, depending on the preprocessing method used, either RF or SVM yield a slightly higher precision.

The average F_1-score of RF is 83%, about 8 percentage points higher than the average F_1-score yielded by SVM. This is due to the much higher recall obtained by RF. The higher F_1-scores imply an overall better performance of RF for all preprocessing techniques. For the *Lemma*, *NegationTagged*, *NegationRemoved* and *Tagged* methods, the difference in F_1-scores obtained by RF and SVM is statistically significant.

The performance results vary depending on which classifier is used, but also based on the applied preprocessing technique. Here, the various preprocessing techniques affect the performance of the two classifiers differently. However, none of the differences in results are statistically significant.

RF performs well when *Lemma*, *NegationTagged* or *NegationTagged + IST* is used. *Lemma* yields the best result for RF, obtaining a precision of 83%, recall of 87% and F_1-score of 85%.

SVM performs best when *IST*, *NegationTagged + IST* or *NegationRemoved + IST* is applied. The method *NegationRemoved + IST* yields the best result for SVM, obtaining a precision of 86%, recall of 74% and F_1-score of 80%. It is interesting to notice that this method increases precision considerably when compared to the *Plain* method, i.e., from 80% to 86%.

TF-IDF 50 yields the lowest performance for both classifiers. This could be explained by the fact that only occasional events in the HRs refer to HAI. Thus, terms used to describe HAI appear rather infrequently and are therefore not represented by the 50 most frequent terms that are used by the *TF-IDF 50* method.

With regard to AUC, RF outperforms SVM for all preprocessing methods. The highest AUC of 0.87 is observed for RF when the *NegationRemoved + IST* preprocessing method is used. RF-*Lemma* that yields the highest performance for RF with regard to recall and precision, obtains an AUC of 0.85, revealing that the classifier is making an informed decision with a good dependence between the features and the class.

It is interesting to point out that RF significantly increases recall compared to SVM. On the other hand, both classifiers perform similarly with regard to precision. Since we want the classifier to ideally maximise both measures, i.e., yield the highest possible F_1-score, the results based on our data suggest that RF is more suitable for the task. The overall best result was yielded by RF-*Lemma*.

5.2 Error analysis

An error analysis of the HRs classified by RF shows that the classification, to some extent, was dependent on the load of infections in the classified HRs. HRs containing HAI were correctly classified if there was a more severe infection, or if the HR contained multiple infections of both types, HAI and CAI. The FNs were more often HAI of a less severe type, or, if the patient had several healthcare episodes following the discovery of HAI, the HAI infection was not elaborated on in the narratives for the subsequent healthcare episodes, and was therefore possibly misjudged by the classifier.

Patients with infectious symptoms such as fever, but without any detectable bacteria, were sometimes misclassified as having HAI. However, the classifier did not strictly classify by infection or no-infection: it was more of a trend in the different groups.

The assumption that CAI may be more often misclassified as HAI than NoINF can be confirmed. For RF-*Lemma*, 12 of the 24 FPs are CAI. Given that there are 25 CAIs in total, 48% of all CAIs are incorrectly classified as HAI. On the contrary, only 12 of 58 total NoINFs (21%) are incorrectly classified as HAI. A similar distribution was observed for all other preprocessing methods for both algorithms.

6 Discussion

6.1 Temporality

For both classifiers and all preprocessing techniques, a greater percentage of CAI than NoINF was misclassified as HAI. This could indicate that the classifiers had difficulties in distinguishing between the cause of infections. The reason for this may lie in the nature of how these infections are described in the HRs, in combination with the approach we use to process these HRs. HRs respectively containing HAI and CAI comprise similar terms, for example, terms that refer to symptoms, such as *feber* (Engl.: fever), or to medication. While being similar in regard to which terms occur in the record, HAI and CAI do differ and can be distinguished based on the temporal context in which the terms appear. This means that, despite the terms in both descriptions being similar, the cause of events and temporality of the events are different. For both RF and SVM, however, our approach does rely solely on terms and their occurrence, which illustrates that the syntactic, semantic or temporal context is not taken into account.

6.2 Comparison to related work

Detecting HAI in EHRs has been studied in previous approaches. However, the effect of negation detection and term reduction on classifier performance has not yet been analysed.

Our experiments suggest that we can achieve results that are in line with those obtained in approaches by other research groups, even though they are not directly comparable due to different datasets and variants of languages used. Benhaddouche et al. obtain their highest recall at 96.64% for the internal and 82.76% for the external test set (Benhaddouche and Benyettou, 2012), while Cohen et al. obtain a maximum recall of 92.6% (Cohen et al., 2004); finally, Iavindrasana et al. obtain a recall of 82.56% (Iavindrasana et al., 2009).

6.3 Limitations

One limitation concerns the fact that the dataset does not represent the real-life distribution of HAI and NoHAI, which corresponds to about 10% HAI and 90% NoHAI. Moreover, our dataset is not balanced but skewed towards the positive class. However, RF has been proposed as appropriate when handling imbalanced data (Dehzangi et al., 2010), so the imbalanced data can be considered a minor limitation compared to the fact that it does not represent real-life distribution.

Another limitation is the fact that our dataset is fairly small. Obtaining more data will hopefully improve the results in a future approach.

The terminology that we use in order to detect ISTs is also a source of limitation. With 2,201 terms, the terminology is comprehensive, containing a number of relevant terms. Yet it was not checked for its consistency regarding the inclusion of inflected forms for the respective terms.

In addition, another limitation of the study is that no inter-annotator agreement was calculated, although two annotators evaluated whether the HRs contained HAIs or not. This was due to the fact that the physicians considered the task to be very difficult. Therefore, they decided to perform the assessment and subsequent annotation together instead of individually.

7 Conclusion

The best result was obtained by Random Forest when lemmatisation was applied as a preprocessing method, yielding a precision of 83%, recall of 87% and F_1-score of 85%. The AUC of 0.85 implies that the classifier is making an informed decision with a good dependence between the features and the class.

Applying Random Forest to our small and imbalanced dataset proved to be a good choice. With regard to recall, the overall performance and AUC, Random Forest outperforms Support Vector Machines for all preprocessing techniques and can therefore be considered more applicable to the task.

Preprocessing affected the performance of Support Vector Machines, especially with regard to precision, slightly more so than the performance of Random Forest. However, none of the preprocessing methods led to statistically significant improvements in performance results. It would therefore be interesting to explore other combinations of preprocessing methods, to see if any of these will improve the results significantly. An example would be to combine the *Lemma* and *NegationRemoved + IST* methods, each of which yielded promising results in this study.

Distinguishing between records that contained healthcare-associated and community-associated infections respectively was difficult for the classifier, since terms describing each kind of infection are similar. A future approach could focus on incorporating temporality to address this.

Acknowledgment

The authors would like to thank Hideyuki Tanushi for preparing the data and Maria Skeppstedt for good advice on using NegEx.

References

Klaus-Peter Adlassnig, Alexander Blacky, and Walter Koller. 2009. Artificial-Intelligence-Based Hospital-Acquired Infection Control. *Strategy for the Future of Health, Studies in Health Technology and Informatics*, 149:103–110.

Djamila Benhaddouche and Abdelkader Benyettou. 2012. Control of Nosocomial Infections by Data Mining. *World Applied Programming*, 2(4):216–219.

Alexander Blacky, Harald Mandl, Klaus-Peter Adlassnig, and Walter Koller. 2011. Fully Automated Surveillance of Healthcare-Associated Infections with MONI-ICU – A Breakthrough in Clinical Infection Surveillance. *Applied Clinical Informatics*, 2(3):365–372.

Constantinos Boulis and Mari Ostendorf. 2005. Text Classification by Augmenting the Bag-Of-Words Representation with Redundancy-Compensated Bigrams. In *International Workshop in Feature Selection in Data Mining*, pages 9–16. Citeseer.

Aodhán S. Breathnach. 2009. Nosocomial Infections. *Medicine*, 37(10):557–561.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45:5–32.

CDC. 2013. Healthcare-Associated Infections (HAI).

Gilles Cohen, Mélanie Hilario, Hugo Sax, Stéphane Hugonnet, Christian Pellegrinib, and Antoine Geissbuhler. 2004. An Application of One-class Support Vector Machines to Nosocomial Infection Detection. In *In Proc. of Medical Informatics*, pages 716–720.

Mita K. Dalal and Mukesh A. Zaveri. 2011. Automatic Text Classification: A Technical Review. *International Journal of Computer Applications*, 28(2):37–40.

Hercules Dalianis, Martin Hassel, and Sumithra Velupillai. 2009. The Stockholm EPR Corpus - Characteristics and Some Initial Findings. In *14th International Symposium for Health Information Management Research – ISHIMR 2009*.

Abdollah Dehzangi, Somnuk Phon-Amnuaisuk, and Omid Dehzangi. 2010. Using Random Forest for Protein Fold Prediction Problem : An Empirical Study. *Information Science and Engineering*, 26:1941–1956.

Eduard Dragut, Fang Fang, Prasad Sistla, Clement Yu, and Weiyi Meng. 2009. Stop Word and Related Problems in Web Interface Integration. In *Proceedings of the VLDB Endowment*, pages 349–360.

Georges Ducel, Jacques Fabry, and Lindsay Nicolle, editors. 2002. *Prevention of Hospital Acquired Infections: A Practical Guide*. WHO, 2 edition.

R. Scott Evans, Rouett H. Abouzelof, Caroline W. Taylor, Vickie Anderson, Sharon Sumner, Sharon Soutter, Ruth Kleckner, and James F. Lloyd. 2009. Computer Surveillance of Hospital-Acquired Infections: A 25 year Update. In *AMIA Annual Symposium Proceedings*, volume 2009, page 178. American Medical Informatics Association.

Christopher Fox. 1989. A stop list for general text. In *ACM SIGIR Forum*, volume 24, pages 19–21. ACM.

Robert E. Hoyt, Nora Bailey, and Ann Yoshihashi. 2012. *Health Informatics: Practical Guide For Healthcare And Information Technology Professionals*. 5 edition.

Jimison Iavindrasana, Gilles Cohen, Adrien Depeursinge, Rodolphe Meyer, and Antoine Geissbuhler. 2009. Towards an Automated Nosocomial Infection Case Reporting-Framework to Build a Computer-Aided Detection of Nosocomial Infection. In *Proceedings of the Second International Conference on Health Informatics, HEALTHINF 2009*, pages 317–322.

Nathalie Japkowicz and Mohak Shah. 2011. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

László A. Jeni, Jeffrey F. Cohn, and Fernando De La Torre. 2013. Facing Imbalanced Data Recommendations for the Use of Performance Metrics. In *2013 Humane Association Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 245–251.

Bart Jongejan and Dorte Haltrup. 2013. The CST Lemmatiser.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

William S. Noble. 2006. What is a Support Vector Machine? *Nature Biotechnology*, 24(12):1565–1567.

Yanjun Qi. 2012. Random Forest for Bioinformatics. In *Ensemble Machine Learning*, pages 307–323. Springer.

Magnus Sahlgren. 2005. an introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, pages 1–9.

Kansheng Shi, Jie He, et al. 2011. Efficient text classification method based on improved term reduction and term weighting. *The Journal of China Universities of Posts and Telecommunications*, 18, Supplement 1(0):131–135.

Bengt Sigurd. 1991. *Språk och språkforskning*. Studentlitteratur, (In Swedish).

Maria Skeppstedt. 2011. Negation Detection in Swedish Clinical Text: An Adaption of NegEx to Swedish. *Journal of Biomedical Semantics*, 2(Suppl 3):S3.

Thomas G Tape. 2006. Interpreting Diagnostic Tests. *University of Nebraska Medical Center, http://gim.unmc.edu/dxtests/ROC3.htm*.

Cornelis Joost Van Rijsbergen. 1979. Information Retrieval. Dept. of Computer Science, University of Glasgow. *http://www.dcs.gla.ac.uk/Keith/Preface.html*.