MeRC

McMaster eBusiness Research Centre

Automatic Message Triage: A Decision Support System for Patient-Provider Messages

By

Amir Tavasoli and Norm Archer

McMaster eBusiness Research Centre (MeRC)

WORKING PAPER No. 42
August 2012

McMaster University

AUTOMATIC MESSAGE TRIAGE: A DECISION SUPPORT SYSTEM FOR PATIENT-PROVIDER MESSAGES

By

Amir Tavasoli and Norm Archer

MeRC Working Paper #42
August 2012

©McMaster eBusiness Research Centre (MeRC)
DeGroote School of Business
McMaster University
Hamilton, Ontario, L8S 4M4
Canada

tavasoa@mcmaster.ca
archer@mcmaster.ca

ABSTRACT

Background: Email communication between patients and healthcare providers is gaining popularity. However, healthcare providers are concerned about being inundated with patient messages and their inability to respond to messages in a timely manner. This work provides a text mining decision support system to overcome some of the challenges presented by email communication between patients and healthcare providers.

Method: A decision support system based on text mining algorithms was developed and tested to triage real world email messages into medium and highly urgent messages that are routed to health provider staff, or low urgency messages that could be routed to an automated response system, responding to the messages in a timely and appropriate way.

Results: Due to the length of email messages, feature reduction algorithms are inadequate in this context. Therefore, in this work, several different classifiers were combined and tailored to build a high performance classifier that supports this type of classification. The system was tested and proved to perform well with real-world patient messages that were exchanged with healthcare providers during a hypertension management study.

Authors' contributions

This work was developed based on a thesis titled "Automatic Message Triage" submitted to School of Graduate Studies at McMaster University by the primary author in partial fulfillment of the requirements for the M.Sc. Degree in Computer Science. The second author of this work supervised the development of the thesis and this article.

Acknowledgements

This research was supported through a grant from the Natural Sciences and Engineering Council of Canada. The authors would like to acknowledge the cooperation of the MyOSCAR research team in the Department of Family Medicine at McMaster University, including Dr. David Chan, Christine Rodrigues, and Dr. Lisa Dolovich, for providing the patient messages, triage levels, help in triaging the messages. Their help is greatly appreciated.

Keywords: Text mining, classification, triage, personal health records, information systems, decision support systems

INTRODUCTION

Electronic personal health record (ePHR) systems enable patients to create, access, and manage their health and medical records over the Internet[1,2]. These systems are supporting a change to a patient-centered focus, causing a revolution in healthcare delivery strategy[1,2]. Ross and Lin[3] suggest that the use of ePHR systems has positive effects on patient-provider communication. Most ePHR systems enable patients to send messages to their providers through a mechanism like secure messaging[4] or email, and more patients are beginning to use web messaging and email to communicate with their healthcare providers[5,6]. This can reduce errors in communication, increase the productivity of healthcare providers, enhance patient access to healthcare providers, and boost patient satisfaction[5,6].

Despite these benefits, patient-provider communication has caused many concerns, including the potential for additional unpaid work by providers[5,6]. This may result in overload when healthcare providers are inundated by non-urgent patient messages, while some messages needing an emergency response may be neglected[6]. In order to tackle these issues some researchers have used triaging systems where messages are sent to a central dispatch center. Here, staff members can triage messages according to their urgency, and send urgent messages on to healthcare providers[7].

The advent of the Internet and the resulting explosion of information worldwide[8] has led to the development of automated text mining techniques to help organize this vast ocean of information into various categories[9]. An important and relevant application of text classification is categorizing text documents such as patient email messages, due to their significance for healthcare providers. Based on a special journal issue concerning the use of text and data mining in facilitating healthcare and ePHRs[10] this work proposes a decision support system (DSS), which uses text mining techniques to triage patient email messages automatically. This can be a great help to healthcare practitioners in increasing the speed of response to urgent messages and also to handle non-urgent patient email messages in a timely manner.

This paper begins with a review of literature related to text mining and classification algorithms and then discusses the design issues of decision support for message triaging. This is followed by a review of the test results, and finally a discussion of appropriate algorithm choices, related design decisions, and future trends for automatic DSS message triage.

LITERATURE REVIEW

Patient-Provider Communication

Patient-provider communication plays an important role in patient healthcare and quality of life[11-13]. Good communication helps patients to better understand relevant medical information[6]; for example, instructions they need to follow to improve their healthcare, and facilitating patient motivation to make behavioural changes to improve their health through diet or exercise[13]. Moreover, effective communication increases patient satisfaction with care[6,14]. Communication is traditionally face-to-face, but electronic communication between patients and healthcare providers has become popular due to the advent of the Internet[6]. Electronic communication increases access to healthcare providers[6,15], leads to better health quality[6,15,16], and makes patients feel more comfortable and satisfied[6,17,18] Moreover, it makes it more convenient for them to ask pertinent questions of their healthcare providers[6,19]. Electronic communication also eliminates the need for unnecessary visits or phone calls by patients and increases the pace of healthcare delivery[18]. In some cases email has been used for remote consultations[20,21].

Email communication with patients has been adopted slowly[6] because of certain concerns of patients and healthcare providers, including added workload for healthcare providers, use of email for improper purposes[6,22-25], and a lack of compensation for the resulting time demands on healthcare providers[6,25,26].

In order to overcome these challenges there have been several suggestions in the literature. For example, Ye et. Al[6] suggest structures like limiting the number of characters in patient emails to reduce the amount of email to be read by healthcare providers. Another solution is to use trained staff to triage messages, responding to low priority messages directly, but sending higher priority messages to the appropriate healthcare providers[2,7,27].

The objective of this work was to design a DSS system that uses text mining to automatically triage messages sent from patients and to potentially manage the patient email challenges mentioned. By automatically classifying patient messages, it can pass along medium priority messages for handling by staff, and it can pass urgent messages directly to the appropriate physician for immediate response. It also opens doors to the design of automated systems that are able to respond directly to low priority requests for information. This builds on experience with text mining in other healthcare related areas such as document labelling[28].

Text Classification

Text classification, also, known as text categorization, is a two phase process, where the first phase involves the construction of a model from a training set of text documents and the second phase uses the model to classify new unknown text documents[9,29]. In order to build a model from training documents, the first step is to break text into words or "tokens" (known as "tokenization")[9]. An appropriate data representation for the text documents must be chosen, with the most common data representation known as the spreadsheet model or "bag-of-words"[9]. In the spreadsheet model the text document is represented as a "feature vector" consisting of document token frequencies. Examples are depicted in Table 1.

Table 1 – Spreadsheet Representation of Text Documents

		Tokens		
		CN Tower	*Celebration*	*Toronto*
Documents	doc_1	1	0	5
	doc_2	0	2	1
	doc_3	1	0	1
	doc_4	2	4	0

Another well-known technique in text representation is the multiword feature[9] or n-grams[8,30] in which groups of tokens (words or characters) that are related to each other (occur together frequently such as 'CN Tower' in Table I), are treated as one token in the document vector. When the right data representation techniques have been chosen, these approaches may result in a huge number of tokens in each document. To filter these tokens and choose only the most important ones, algorithms for feature reduction have been developed. For example, removing stop words, like "to" or "for", from a document is a commonly used feature reduction technique[9,29].

There are several "classification algorithms" that can be used to build a model based on a set of text documents and then to use the model to classify a new and unknown text document[9,29]. Several of these classifiers are discussed in the following. Their applications in this research are described in the Methods section.

K Nearest Neighbour Classifier

The K Nearest Neighbour (KNN) classification method is based on similarities between documents[9]. After choosing the right data representation, followed by tokenization and feature reduction, KNN algorithms save the resulting training feature vectors for future use[8]. The KNN classifier converts any new document it is given into a feature vector and calculates the distance between the new vector and the previously calculated feature vectors. The k nearest neighbours of the new document are then selected and the new document is classified into the most frequent category appearing among its k nearest neighbours[8,9]. The similarity measure algorithm to calculate these distances is most commonly the standard Euclidian distance between vectors[8,31]. There are other similarity measure algorithms like the cosine distance between vectors, based on the angle between two vectors[8,31]. KNN performance is among the best of the algorithms used in text classification[29].

Language Model Classifier

The Language Model (LM) classifier has been widely used in many other areas like speech recognition[32,33]. A LM classifier uses a comparison of the n-grams in the new unknown document with those in the training documents of a certain category[8,31]. If an n-gram has appeared in the training corpus it will have a higher chance of belonging to that category of n-gram in the training corpus[31].

Naïve Bayes Classifier

The Naive Bayes (NB) classifier categorizes text documents based on the Bayes theorem of probability[29]. After the initial stages, the algorithm calculates the probability that a new document belongs to each category, and classifies the new document in the class with the highest probability. The NB classifier is based on the rigorous assumption that all features are probabilistically independent, so this classifier is known as "naive"[29]. Because of the naive assumption, this classifier does not perform as well as many other classifiers, and is mostly used as a classifier in comparing classifier performance.

Term Frequency-Inverse Document Frequency Classifier

The Term Frequency (TF) – Inverse Document Frequency (IDF) classifier is based on the TF-IDF rule[8,31]. TF represents the frequency of a certain token in a document, and IDF represents the inverse of the number of categories in which this token has occurred[9,29]. The product of TF and IDF for the token is a weight that represents the importance of that token in classification[9,29].

Multiple Classifiers

In everyday life if several experts give their suggestions on a subject it is more likely to result in a better decision than a decision based on single expert judgment. Results from a combined classifier are also more likely to be precise than a single classifier. Several approaches can be used in combining classifiers. The simplest form of combining classifier results is known as voting. In this method the new document is classified into the category which most individual classifiers have voted for[34]. Another common method for classifier combination is known as adaptive classifier combination (ACC). ACC combines the results of various classifiers in order to get better results[34].

SYSTEM DESIGN

In order to overcome the issues mentioned in email communication between patients and healthcare providers, the authors have designed a text mining DSS to automatically categorize messages according to their importance, to help nurses and physicians to manage patient email messages. System design is depicted in Figure 1.

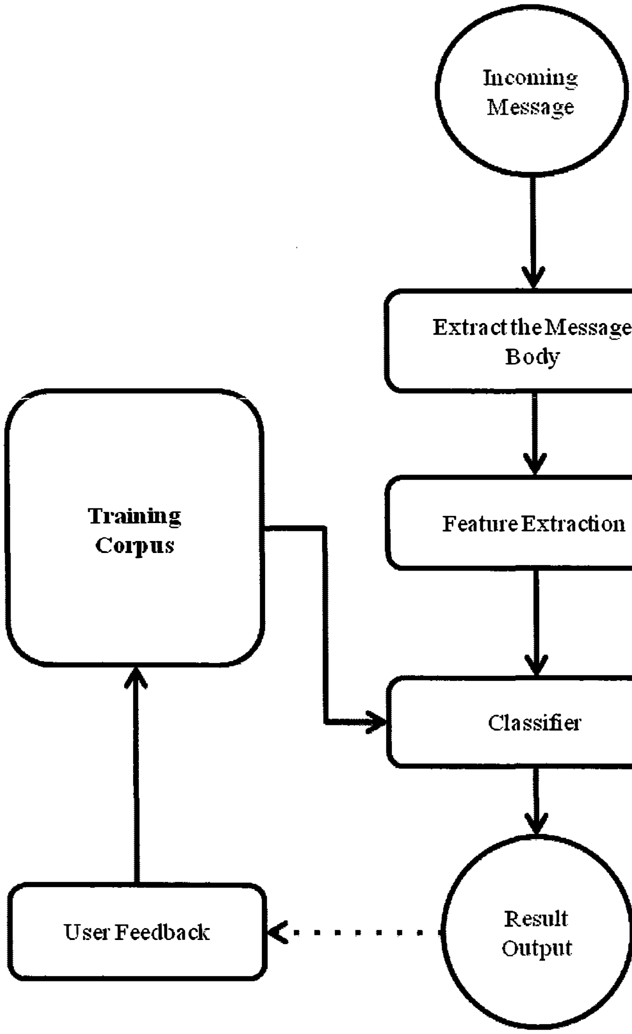

Figure 1 – Overall design of automatic message triage system

The system extracts the message body from each new incoming message. The feature extraction algorithm converts the new message into a feature vector, which is readable by the classifier algorithm. Finally, the classifier shows the results to the user and, if the user provides feedback, the message is added to the training corpus to increase the future precision of the classifier. If there is no feedback from the user, the message is just assigned a triage level and will not be used for system improvement. For optimal design of the DSS, the best available feature extraction and classifier algorithm for this environment must be chosen. Choosing these algorithms is discussed in the next section.

METHOD

Classification Algorithms

For the implementation of classification algorithms there are several open-source customizable packages, including GATE[35], WEKA[36], and LingPipe[37]. LingPipe was chosen for this work since it is flexible and relatively easy to use. The DSS was implemented using the Java language and the LingPipe library.

Linear classifiers were chosen for this work. The performance of some non-linear classifiers like neural networks has no significant advantage over linear classifiers in text categorization[29,38,39]. Also, non-linear classification methods, even if they generate good results, lack clarity as they are black box solutions, limiting user access to the underlying model[40]. The latter is very useful in our work, since it allows checking, evaluating, and improving the model, resulting in potential useful extensions to the DSS. Linear classifiers have the useful feature of learning and adapting to their environment quickly[29]. For example, users can change triage levels at any time, supported by a rapid transition and adaptation of the system to its new levels.

LingPipe supports several linear classification methods that can be used in the system[31]. The KNN, LM, and TF-IDF classifiers were chosen for testing because they are commonly used and have good performance.

Because KNN, LM, and NB classifiers have a standard implementation in most text mining packages including LingPipe, only the implementation details of TF-IDF classifiers is discussed here.

Indo-European Tokenizer

The Indo-European tokenizer for extracting tokens from text documents[31] recognizes patterns like sequence of numbers, letters, and etc[31]. An advantage of this tokenizer is that it supports various languages like English, French or Spanish.

Feature Reduction Algorithms

Email messages are usually brief[6,41-43], so any reduction in the number of features can reduce classifier performance. Consequently, for this work no specific feature reduction or selection algorithm was used. However, the TF-IDF classifier was tested as a special kind of feature selection.

Term Frequency-Inverse Document Frequency Classifier

TF-IDF is mostly known as a feature reduction technique, but LingPipe's implementation has been tweaked so it can be used as a classifier[31]. In the training phase this classifier uses the Indo-European tokenizer to convert documents to feature vectors. These feature vectors are smoothed using the TF-IDF algorithm before they are saved. LingPipe calculates the TF-IDF weight of each of these tokens by multiplying TF and IDF and saves them in the training phase[31]. Therefore, a token that has been exclusively used in one document and in one category will have a higher weight than other tokens for classification. In the classification phase, the cosine distance of a newly given vector with the current training vectors in each category is calculated, and the new document classified into the category with which its vector has the best fit[31].

As discussed, feature reduction techniques do not add to classifier performance, because of the small number of features due to the short length of email messages. In fact, testing in this environment showed that use of TF-IDF feature reduction on the LM classifier actually reduced its precision by 24%. This implementation of TF-IDF performed relatively well because none of the features were omitted from consideration.

Table 2 summarizes the technical specifications of each classification method implementation.

Table 2 – Summary of Classification Algorithms Implementation Details

Classifier Type	Data Model	Tokenizer Algorithm
KNN classifier	Bag-of-words	Indo-European Tokenizer
LM classifier	n-gram based on characters	No Tokenizer
NB classifier	Bag-of-words	Indo-European Tokenizer
TF-IDF Classifier	Bag-of-words	Indo-European Tokenizer

Combination Algorithms

In this work simple voting, ACC, and a new ACC derivation were used to combine classifiers. All of the combination classifiers combine the results from all of the four individual classifiers mentioned. Implementation details of simple voting have not been included because they used standard implementations.

Adaptive Classifier Combination

The implementation of ACC is mostly inspired by Li and Jain[34]. Steps in this method are:

1. Find the k nearest neighbourhood of the newly given document using standard neighbourhood algorithms like Euclidean or cosine distance methods.

2. The given document is classified using all of the 4 individual classifiers mentioned.

3. For all categories a measure called *Acc* is calculated by multiplying the cosine distance of the newly given document and the located neighbourhood by the probability that the new document belongs to that category, as calculated by the classifier.

4. The document is classified in the category that has the highest *Acc* measure.

New Neighbourhood Method

In order to increase the performance of combination methods in this environment, the authors propose an innovative way to locate neighbours in the ACC method. As mentioned, there are a small number of features in each document. Moreover, there are not many similarities between documents in each triage category. To overcome this problem a new soft cosine measure was used for finding the neighbourhood in the first step of the ACC calculation:

1. Use the cosine measure to find the *k* nearest neighbours of the given document.

2. If fewer than three neighbourhoods are found for the document, the calculation is complete.

3. If there are more than two neighbourhoods, then start with the neighbourhood that is least similar to the new document, and check to see if the cosine distance between the two documents is less than 0.7 (i.e. the degree between feature vectors is less than 45 degrees or the documents are roughly less than half similar). If this is the case, then delete that neighbourhood and continue till there are two neighbourhoods left or the cosine distance is more than 0.7. Using this method, only "good" neighbourhoods that are at least half similar to the newly given document will be in its neighbourhood, and the precision of the ACC method will increase.

DATA COLLECTION

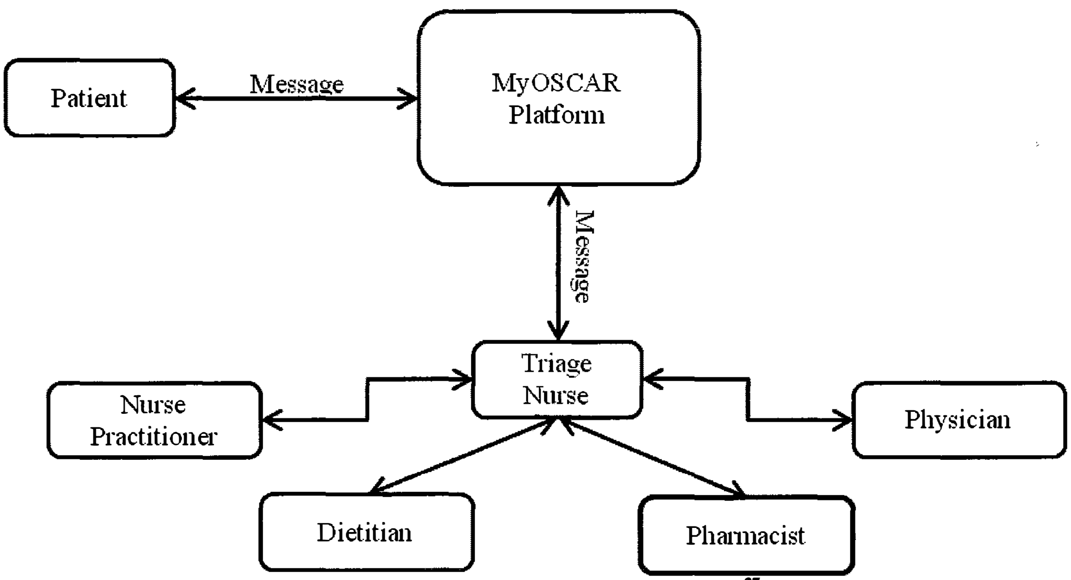

Figure 2 – MyBP Email Messaging System[27]

Messages used for this study were obtained from the My Blood Pressure (MyBP) study[27], which evaluated the effects of ePHRs on patient hypertension self-management. In this study, patients had access to a customized version of MyOSCAR, which is an online, open source, secure electronic personal health record (ePHR) system[44]. In the MyBP study, patients could record their blood pressure, see their blood pressure charts, drug prescriptions, and contact their healthcare providers through a secure email system[27]. Their email messages were used to design and test the automatic triage system developed in the current work. In the MyBP study, a triage person read and redirected incoming messages to the appropriate healthcare providers (pharmacists, dieticians, nurse practitioners, and primary care physicians). Figure 2 shows the email messaging system of the MyBP study.

During the study 1460 email messages where exchanged between patients and healthcare providers[27]. 1296 of these were general email messages, (e.g. reminders), sent by the nurse as reminders to patients. The remaining 164 email messages were used in designing and testing the automatic message triage DSS. These email messages could concern clinical subjects or technical issues. Based on the subject and importance they were answered in a time ranging from immediate up to at most 72 hours. All the email messages were responded to, even if the response was a simple thank you. Table 3 shows the email message categories and their response times.

Table 3 – My Blood Pressure Study Email Message Categories and their Response Times

Type		Timing	Category
Clinical	Personal	Immediate	A
		Within 24 hours	B
		Within 72 hours	C
	Generic	Within 72 hours	D
Technical (IT) / Data collection		Within 72 hours	E
Duplicate Messages			F

DATA MANAGEMENT

The current study was approved by the McMaster Faculty of Health Sciences/Hamilton Health Sciences Research Ethics Board. To prepare the MyBP messages for this study, the bodies of all the messages were extracted from the 1460 available raw xml messages. All header data including senders, receivers, etc. were deleted. Any personally identifiable data was also deleted from all of the email messages to protect patient privacy. Because there were a large number of messages, a text mining program called GATE[35] was used to detect and delete names and other personal data in the email messages. The final message set was reviewed by the authors to ensure that all personal data had been removed. The remaining 164 email messages were grouped into four triage levels based on their required response time, with the help of two nurses who worked with the MyBP study team. To ensure the correctness of the triage levels, the nurses triaged the messages independently. The few disagreements in assigned triage levels were resolved by the authors before the resulting triage assignments were used. The triage levels are shown in Table 4.

Table 4 – Triage Levels for Email Messages

Triage Level	Description
Level 0	*Immediate action* – probable emergency. *Includes message category A.*
Level 1	*Immediate action* – to be handled by the triage person. *Includes message category A.*
Level 2	*Response within 24 hours* – to be handled by triage person. *Includes message category B.* Examples: Patient appointment times; anything to do with non-urgent elevated / potentially blood pressure readings
Level 3	*Response within 72 hours* (since the aim is that all messages will be addressed within 3 days). *Includes message categories C, D and E.* Example *C*: Issues with personal blood pressure monitors Examples *D*: Medication changes to be updated; updates to other online personal health record Examples *E*: Issues accessing different aspects / components of online personal health record; difficulties with survey completion; lost passwords; etc
Level 4	*Duplicate message* – not used for classification

The messages were categorized, based on the response time needed for the most urgent issue and not based on the time needed to provide a solution to all the problems if the message addressed more than one issue. Other similar studies have shown that most email messages address only one issue[6,43].

Because there were only 28 patients in the MyBP study and the length of the study was relatively short (1 year), there were no level 0 messages and only one level 1 message, in agreement with other similar studies[15]. Therefore 20 additional messages were simulated by the authors for each of triage levels 0 and 1. Table 5 shows the final number of messages included in the training corpus for each triage level. Only email messages with triage levels 0 to 3 were used for testing and training, because level 4 messages were duplicates of other messages and could be checked and deleted by the automatic triaging system.

Table 5 – Number of Training Messages in each Triage Level

Triage Level	Number of Messages
Level 0	20
Level 1	21
Level 2	19
Level 3	69
Level 4	74

There is no way to estimate the appropriate size of a corpus for training and testing a text mining algorithm[9]. In the real world it is common for the number of messages in each category for training and testing to be different[9]. The only way to check whether there are enough messages in each category is to check the performance of the system. Since the final results from the system that was designed were satisfactory, it can be argued that there were enough messages in each of the sample sets. A related issue is that differing numbers of messages in each triage category may increase the likelihood of bias in the classifier toward a certain category[8], but it is difficult to overcome such potential bias without eliminating valuable training messages. Figure 33 shows the process of data preparation.

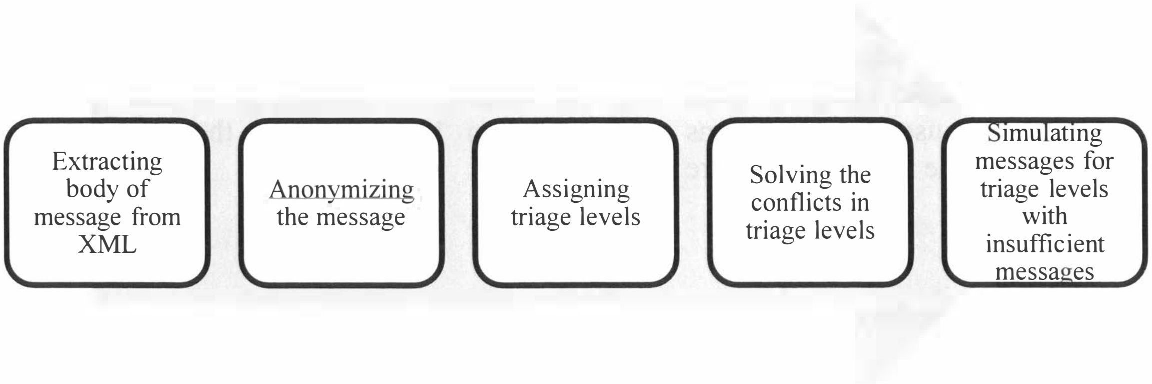

Figure 3 - Data Preparation Process

BENCHMARK CONDITIONS

To choose the most suitable classification algorithm, all of the classification algorithms were tested on the messages. Eighty percent of the messages were used for training and the other twenty percent were used to test the algorithms. Each of the algorithms ran 100 times and in each run the documents were assigned randomly for training and testing to ensure that performance measures were not biased toward a specific set of training and test messages. Performance results did not show any significant changes above 100 runs.

CLASSIFIER PERFORMANCE

Several measures have been used in the literature to compare classifier performance, with the choice of measure being based on the application context. Three different measures were used in this study. The first was error rate[9], where:

$$erate = \frac{number\ of\ errors}{number\ of\ documents}$$

The second was a measure known as "critical error" or *Cerr*. In the context of this application, if an email message with a lower priority is erroneously classified as a message with a higher priority (e.g. triage level 0 or 1), this is not a critical error. However, if an email message with a higher priority is classified as a lower priority message, this is a critical error, since it can be dangerous to patient health. If level 2 or 3 messages are misplaced with one another there will not be serious consequences for the patient and consequently this type of error is not considered to be critical. For the purposes of this study, the critical measure is defined as the number of level 0 (absolute emergency) email messages erroneously triaged into levels 1, 2, or 3, plus the number of level 1 messages categorized as level 2 or 3 messages, all divided by the number of level 0 and level 1 messages.

$$Cerr = \frac{num\ of\ level0\ msgs\ in\ other\ levels + num\ of\ level1\ messages\ in\ level2\ or\ 3}{count\ of\ level0\ msgs + count\ of\ level1\ msgs}$$

Precision and recall were, also, used to measure classifier performance. Precision is the percentage of documents which are properly classified[9]. Recall is the number of correct classifications in a specific category relative to the number of documents in that specific category[9]. Finally, because the algorithms were each run 100 times, all the performance measures mentioned in the results section are averaged.

RESULTS

This section discuss the results of different algorithms ran using the benchmark conditions discussed.

Individual Classifiers

K Nearest Neighbour Classifier

In LingPipe's KNN classifier's implementation, the number of neighbours (k) and distance algorithm can be changed[31]. In order to choose the best value of k the algorithm was run 100 times for k values between 1 to 20. Figure 4 shows the performance of the KNN classifier with k varying from 1 to 10.

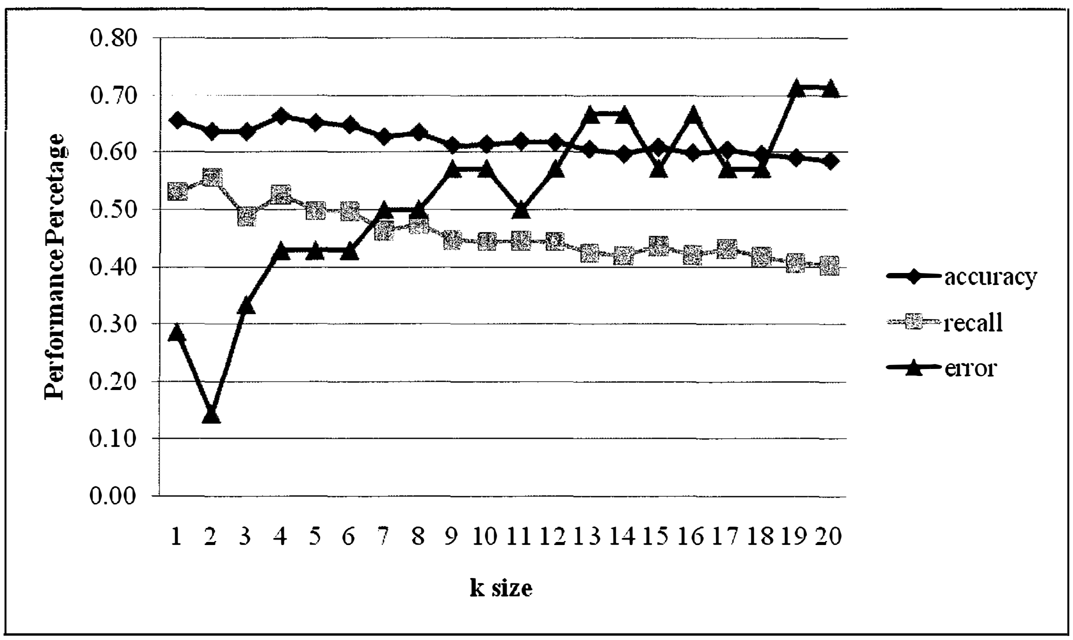

Figure 4 - k and K Nearest Neighbour Classifier Performance Chart

As depicted in Figure 4 when k is 2, *Cerr,* which is critical in this context, reaches its minimum and precision and recall are reasonable. Because there is a relatively small training corpus and the number of features in each vector is small, k=2 was chosen.

Table 6 - Various Distance Algorithm Effects on K Nearest Neighbour Classifier Performance

Distance Function	Precision	Recall	Cerr
Euclidean	0.64	0.56	0.14
Cosine	0.58	0.54	0.00

The next step was to find the appropriate distance algorithm. Euclidean and cosine distance algorithms are the most commonly used. The performance measures of these two algorithms are depicted in Table 6. The Euclidean distance had better overall performance and was chosen as the distance algorithm for the KNN classifier.

Language Model Classifier

For this algorithm the only parameter to be determined was the character base n-gram size[31]. Different n-gram sizes were tested, each for 100 runs. The results are shown in Figure 5.

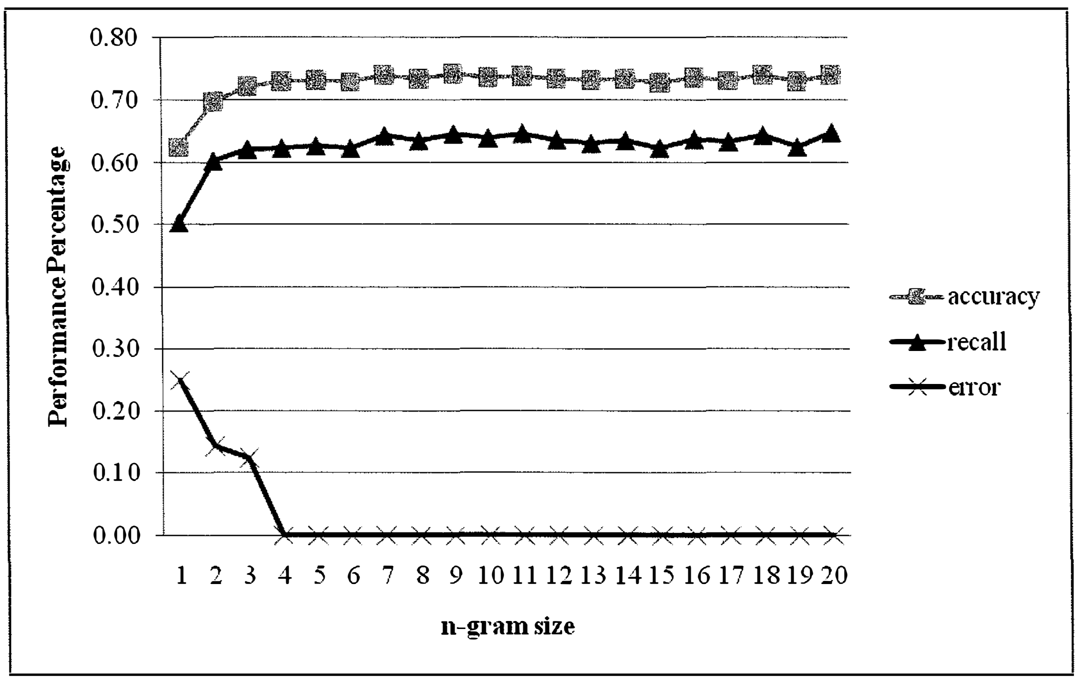

Figure 5 – Language Model Classifier Performance for Various n-gram Sizes

Because n-gram sizes larger than 7 have little effect on classifier performance, 7 was chosen as the optimum n-gram size.

Naïve Bayes Classifier

LingPipe's implementation of the NB classifier does not require any input parameters[31]. Performance measures of this classifier can be located in Table 8.

Term Frequency-Inverse Documentary Frequency Classifier

The TF-IDF classifier implementation in LingPipe does not require any parameter inputs[31]. Performance measures of this classifier can be located in Table 8.

Classifier combinations

Combination algorithms combine the results of all four individual classifiers mentioned and they, also, have been tested on the same benchmark conditions as the single classifiers.

Simple Voting Combination

Simple voting was used as the baseline of comparison between classifier combinations. Performance measures of this classifier can be located in Table 8.

Adaptive Classifier Combination

In the ACC classifier the number of neighbours (k) and the distance algorithm were again chosen. The classifier was tested for different k values for 100 runs each.

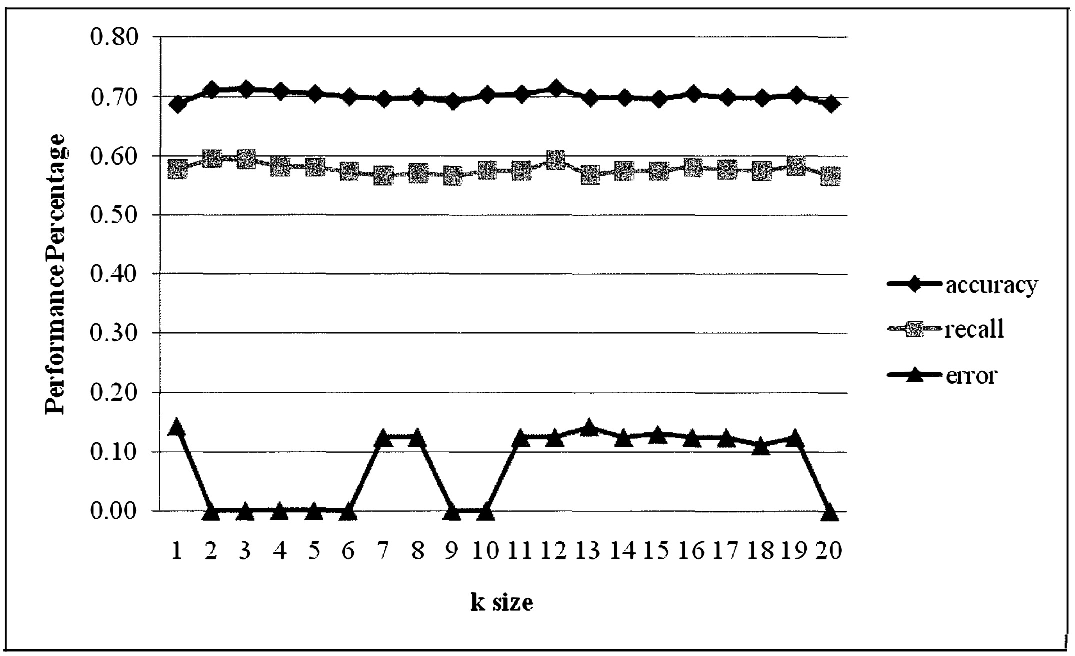

Figure 6 – Adaptive Classifier Combination Classifier for Different *k*s

Based on Figure 6, k=4 seems to be a suitable choice for this algorithm, because it has zero *Cerr* and high precision and recall. For choosing the right distance algorithm two distance algorithms were tested with k=4, as shown previously, with results in Table 7.

Table 7 – Adaptive Classifier Combination Performance for Various Distance Algorithms (with k=4)

Distance	Precision	Recall	Cerr
Euclidean	.71	.58	0.00
Cosine	.73	.65	0.00

Based on the Table 7 results, cosine distance was chosen since it had a positive effect on classifier performance, compared to Euclidean distance.

Adaptive Classifier Combination with New Neighbourhood Algorithm

Because the ACC has its own distance algorithm, only the right k needs to be chosen. Figure 7 shows this algorithm's performance with various k values for 100 runs each.

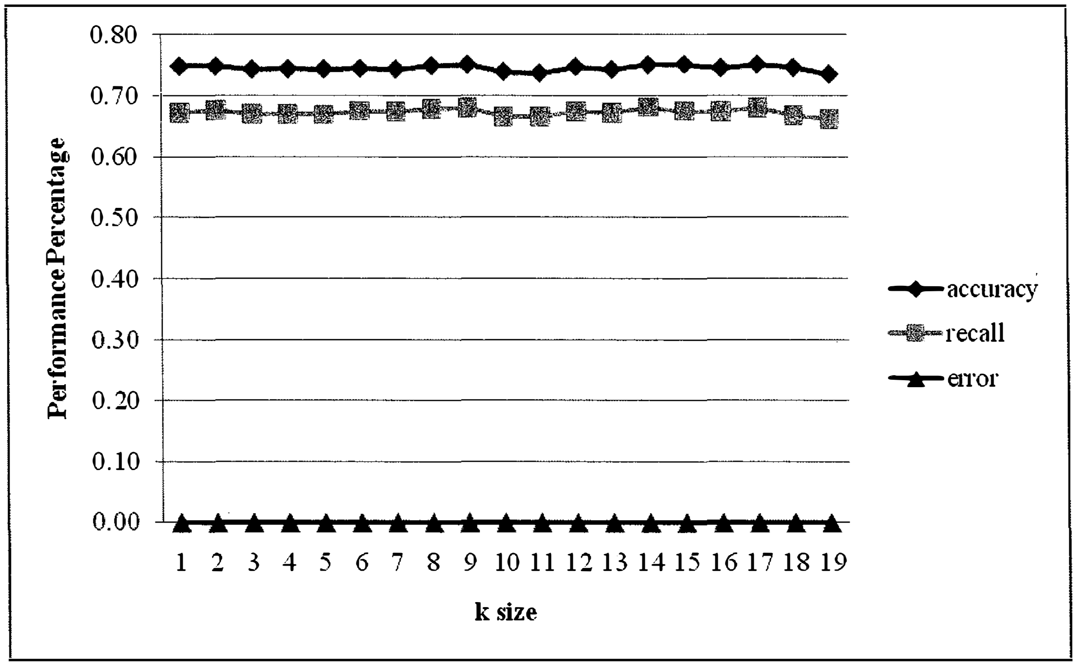

Figure 7 – Adaptive Classifier Combination with new Distance Algorithm with Various *k*s

From Figure 7 results, k=9 was chosen because precision and recall are at their peak and *Cerr* is zero. These results demonstrate that leaving out "bad" neighbourhoods has a positive effect by reducing *Cerr* to zero, which is very important for this context.

Comparison of Classifiers

Table 8 – Classifier Performance Comparison

Algorithm	Precision	Recall	Cerr	Parameters
LM Classifier	.739	.638	.000	n-gram = 7
NB Classifier	.690	.582	.000	N/A
KNN Classifier	.643	.563	.143	k=2 using Euclidean distance

TF-IDF Classifier	.706	.645	.143	N/A
Combination Methods				
Simple Voting Classifier	.720	.621	.000	N/A
ACC Classifier	.733	.648	.000	*k*=4 using Cosine similarity measure
ACC with new distance algorithm	.749	.671	.000	*k*=9

Table 8 compares classifier performance for the different classifiers tested. ACC with the new distance algorithm classifier proved to have the best classifier performance of all the algorithms tested.

DISCUSSION

The results indicate that the ACC with the new distance algorithm classifier is the best choice for triaging email messages sent from patients to healthcare providers. The main reason for this may be the fact that messages in this context are brief, and this classifier uses the "soft" cosine measure to find near neighbours. This may result in finding more refined neighbours, leading to higher performance. These effects can be tested and further enhanced in future improvements of the triage system.

Some published studies disagree with the idea that healthcare providers will be overwhelmed with messages when patients are given the opportunity to communicate with them via email. For example, Liederman et. Al[5] argue that an increase in healthcare provider work performance will compensate for the time they need to spend reading and answering patient email messages. Sitting[42] confirms this by suggesting that the questions that patients now ask via emails used to be asked in telephone or face-to-face meetings. While providing email communication to patients has the potential to replace phone and face-to-face communication[18], there is no guarantee that the fast rate of growth of ePHR systems and patient email communications will not result in the inundation of healthcare providers with email messages. It is likely that automatic triage systems will continue to increase in their precision, with more potential to replace or enhance staff interventions, while at the same time being useful when and where there is a shortage of healthcare professionals.

CONCLUSION AND FUTURE WORK

There are several ways to extend the current system. A useful extension would be to give the system the ability to redirect messages to the intended recipients automatically. There is no reason the system could not be used to triage and redirect any type of patient message sent, including SMS[45]. Another addition to the system would be a program to edit and refine the training corpus when new message classifications are made, resulting in a continuous improvement in the accuracy of the triage process. This would also allow automatic message triage DSS to begin with lower numbers of training messages, and to gain accuracy under the watchful eye of a staff person. Adding the ability for the DSS to answer automatically low priority requests for information could, also, be a very useful system enhancement, particularly in regions where medical personal are scarce.

This study showed that the new neighbourhood algorithm developed for ACC had the best performance of the classifiers tested, for triaging patient email messages. This system was designed specifically for the healthcare field. Similar systems have been developed for other industries, such as Google's priority inbox which sorts emails based on their importance for the user[46]. There are other industries where responses could be generated automatically through a web search engine including, for example, the work by Kovacevic et al.[47] or IBM's Watson[48].

GLOSSARY OF TERMS USED

ACC: Adaptive Classifier Combination

DSS: Decision Support System

ePHR: electronic Personal Health Record

KNN: K Nearest Neighbour

LM: Language Model

NB: Naive Bayes

TF-IDF: Term Frequency - Inverse Document Frequency

REFERENCES

1. Sittig, D.F., *Personal health records on the Internet: a snapshot of the pioneers at the end of the 20th Century.* International Journal of Medical Informatics, 2002. **65**(1): p. 1-6.
2. Archer, N., et al., *Personal health records: A scoping review.* Journal of the American Medical Informatics Association, 2011. **18**: p. 515-522.
3. Ross, S.E. and C.T. Lin, *The effects of promoting patient access to medical records: a review.* J Am Med Inform Assoc, 2003. **10**(2): p. 129-38.
4. Halamka, J.D., K.D. Mandl, and P.C. Tang, *Early experiences with personal health records.* Journal of the American Medical Informatics Association : JAMIA, 2008. **15**(1): p. 1-7.
5. Liederman, E.M., et al., *The impact of patient-physician Web messaging on provider productivity*, Journal of Healthcare Information Management *Vol* 2005. p. 81.
6. Ye, J., et al., *E-mail in patient-provider communication: A systematic review.* Patient Education and Counseling, 2010. **80**(2): p. 266-273.
7. Katz, S.J., et al., *Effect of a Triage based E mail System on Clinic Resource Use and Patient and Physician Satisfaction in Primary Care.* Journal of General Internal Medicine, 2003. **18**(9): p. 736-744.
8. Konchady, M., *Building Search Applications, Lucene, LingPipe, and Gate*2008, Oakton, Virginia: Mustru Publishing. 430.
9. Weiss, S., et al., *Text Mining: Predictive Methods for Analyzing Unstructured Information*2004: Springer, Heidelberg
10. Karsten, H. and H. Suominen, *Mining of clinical and biomedical text and data: editorial of the special issue.* International Journal of Medical Informatics, 2009. **78**(12): p. 786-7.
11. Arora, N.K., *Interacting with cancer patients: the significance of physicians' communication behavior.* Social Science & Medicine, 2003. **57**(5): p. 791-806.
12. Ong, L.M.L., et al., *Doctor-patient communication: a review of the literature.* Social Science & Medicine, 1995. **40**(7): p. 903-918.
13. Street Jr., R.L. and R.M. Epstein, *Key Interpersonal Functions and Health Outcomes*, in *Health behavior and health education: theory, research, and practice*, K. Glanz, B.K. Rimer, and K. Viswanath, Editors. 2008, Jossey-Bass.
14. Dutta-Bergman, M.J., *The relation between health-orientation, provider-patient communication, and satisfaction: an individual-difference approach.* Health Communication, 2005. **18**(3): p. 291-303.
15. Rosen, P. and C.K. Kwoh, *Patient-physician e-mail: an opportunity to transform pediatric health care delivery.* Pediatrics, 2007. **120**(4): p. 701.
16. Katzen, C., M.J. Solan, and A.P. Dicker, *E-mail and oncology: a survey of radiation oncology patients and their attitudes to a new generation of health communication.* Prostate Cancer Prostatic Dis, 2005. **8**(2): p. 189-93.
17. Leong, S.L., et al., *Enhancing doctor-patient communication using email: a pilot study.* The Journal of the American Board of Family Medicine, 2005. **18**(3): p. 180.
18. Wakefield, D.S., et al., *Issues and questions to consider in implementing secure electronic patient-provider web portal communications systems.* International Journal of Medical Informatics, 2010. **79**(7): p. 469-477.

19. Houston, T.K., et al., *Experiences of patients who were early adopters of electronic communication with their physician: satisfaction, benefits, and concerns.* Am J Manag Care, 2004. **10**(9): p. 601-608.
20. Car, J. and A. Sheikh, *Email consultations in health care: 1-scope and effectiveness.* BMJ, 2004. **329**(7463): p. 435.
21. Car, J. and A. Sheikh, *Email consultations in health care: 2-acceptability and safe application.* BMJ, 2004. **329**(7463): p. 439.
22. Kleiner, K.D., et al., *Parent and physician attitudes regarding electronic communication in pediatric practices.* Pediatrics, 2002. **109**(5): p. 740.
23. Hobbs, J., et al., *Opportunities to enhance patient and physician e-mail contact.* International Journal of Medical Informatics, 2003. **70**(1): p. 1-9.
24. Houston, T.K., et al., *Experiences of physicians who frequently use e-mail with patients.* Health Communication, 2003. **15**(4): p. 515-525.
25. Patt, M.R., et al., *Doctors who are using e-mail with their patients: a qualitative exploration.* Journal of Medical Internet Research, 2003. **5**(2).
26. Moyer, C.A., et al., *Bridging the electronic divide: patient and provider perspectives on e-mail communication in primary care.* Am J Manag Care, 2002. **8**(5): p. 427-433.
27. myOSCAR research team, *Patient Self Management Approach for Hypertension Using Personal Electronic Health Records (myBP) (unpublished paper)*, 2009, McMaster University: Hamilton, Ontario.
28. Ginter, F., et al., *Combining hidden Markov models and latent semantic analysis for topic segmentation and labeling: Method and clinical application.* International Journal of Medical Informatics, 2009. **78**(12): p. e1-e6.
29. Feldman, R. and J. Sanger, *The text mining handbook : advanced approaches in analyzing unstructured data*2007, Cambridge ; New York: Cambridge University Press. xii, 410 p.
30. Carpenter, B. *Text Analysis with LingPipe 4.* 2010 [cited 2011; Available from: http://alias-i.com/lingpipe-book/lingpipe-book-0.4.pdf.
31. *LingPipe 4.0 API Documentation (JavaDoc).* 2010; Available from: http://alias-i.com/lingpipe/docs/api/index.html?overview-summary.html.
32. *Language Modeling for Speech Recognition - Microsoft Research.* Available from: http://research.microsoft.com/en-us/projects/language-modeling/default.aspx.
33. Tenenboim, L., B. Shapira, and P. Shoval, *Ontology-Based Classification Of News In An Electronic Newspaper*, in *International Conference "Intelligent Information and Engineering Systems" INFOS 2008*: Varna, Bulgaria.
34. Li, Y.H. and A.K. Jain, *Classification of Text Documents*, in *Proceedings of the 14th International Conference on Pattern Recognition-Volume 2 - Volume 2*1998, IEEE Computer Society. p. 1295.
35. Cunningham, H., et al. *GATE Website - General Architecture fore Text Engineering.* 2010; Available from: www.gate.ac.uk.
36. Witten, I.H. and E. Frank, *Introduction to Weka*, in *Data Mining: Practical Machine Learning Tools and Techniques*2005.
37. Carpenter, B. *LingPipe – a suite of Java libraries for the linguistic analysis of human language.* Available from: http://alias-i.com/lingpipe/.
38. Schütze, H., D.A. Hull, and J.O. Pedersen. *A comparison of classifiers and document representations for the routing problem.* 1995. ACM.

39. Wiener, E., J.O. Pedersen, and A.S. Weigend. *A neural network approach to topic spotting*. 1995. Las Vegas, NV, USA: Univ. of Nevada.
40. Bart, B., et al., *Special issue on white box nonlinear prediction models.* IEEE Transactions on Neural Networks, 2010. **21**(3): p. 527-527.
41. Roter, D.L., et al., *Can e-mail messages between patients and physicians be patient-centered?* Health Communication, 2008. **23**(1): p. 80-86.
42. Sittig, D.F., *Results of a content analysis of electronic messages(email) sent between patients and their physicians.* Bmc Medical Informatics and Decision Making, 2003. **3**(1): p. 11.
43. White, C.B., et al., *A content analysis of e-mail communication between patients and their providers: patients get the message.* Journal of the American Medical Informatics Association, 2004. **11**(4): p. 260.
44. *myOSCAR Manual.* 2010; Available from: http://myoscar.org/users-manual-1.
45. Archer, N., *Smart Interactions for Health Self Management.* Invited Presentation at CASCON Conference, November 3, 2010.
46. Google Research Team. *Google Priority Inbox.* 2010; Available from: http://mail.google.com/mail/help/intl/en/priority-inbox.html.
47. Kovacevic, M., J.Y. Nie, and C. Davidson, *Providing Answers to Questions from Automatically Collected Web Pages for Intelligent Decision Making in the Construction Sector.* Journal of Computing in Civil Engineering, 2008. **22**: p. 3.
48. IBM Research Team. *IBM Watson.* 2011; Available from: http://www-03.ibm.com/innovation/us/watson/index.html.

McMaster University
1280 Main St. W. DSB A202
Hamilton, ON
L8S 4M4

Tel: 905-525-9140 ext. 23956
Fax: 905-521-8995
Email: ebusiness@mcmaster.ca
Web: http://merc.mcmaster.ca