SMS Communication and Announcement Classification in Managed Learning Environments

Ross Clement[1], Mark Baldwin[1], Clive Vassell[2] and Nadia Amin[2]

[1]Harrow School of Computer Science, University of Westminster, Northwick Park, Harrow, UK

[2]Harrow Business School, University of Westminster, Northwick Park, Harrow, UK

Keywords. Managed learning environments, Text classification.

Abstract. A prototype system for sending SMS text messages to students telling them about announcements has been designed and partially implemented. Experiments have been performed to test whether automatic text classification can be used to decide which announcements posted by tutors are urgent and that a SMS text message should be sent informing students. The accuracy of a naive Bayes classifier is not sufficient in itself to decide this, but a flexible classifier and the ability of tutors to override its decisions has promise. How the system would be used would depend on management policies concerning the effects of classification errors.

1 Introduction

In this paper we investigate using machine learning based text classification in managed learning environments. In particular, we report experiments designed to evaluate the feasibility of using text classification to decide which announcements made by tutors on a managed learning environment should be sent to students as SMS text messages.

There has been considerable research into the use of Managed (and Online) Learning Environments [7] in higher education. These allow the publication of course materials, creation of interactive revision aids, and also communication between people enrolled in courses and modules. The authors of this paper have noticed that student participation in MLE mediated communication was initially good, but rapidly declined over time. Perhaps because the novelty factor wore off. Beasley & Smyth [1] investigated the real-world usage of MLEs and found a number of problems, including that even students who found the learning environment valuable did not interact with them properly. Research such as that of Ubon and Kimble [9] investigates the participation of students in MLEs and "online learning communities" from a human viewpoint, investigating what they call "social presence" of tutors and

Clement R., Baldwin M., Vassell C. and Amin N. (2005).
SMS Communication and Announcement Classification in Managed Learning Environments.
In *Proceedings of the 1st International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces*, pages 116-124
DOI: 10.5220/0001422201160124

students. We take a much simpler approach to addressing the problems of students failing to make use of the materials posted on MLEs. If students are occasionally, but not too frequently reminded of the existence of the MLE by text message, then they may be more likely to make better use of the MLE.
Mobile phones are being increasingly important vehicles for accessing online services of various types. The authors believe that the mobile phone will become the vehicle of choice for conducting online transactions. Mobile phones are frequently carried by students at all times, and therefore we believe that material sent to phones has the highest probability of being read by students in a short time span, including in comparison to email messages. The sending of SMS messages would also mean that a data "push" aspect is added to the MLE environment, rather than the current "pull" scenario, where the student will not receive information from the MLE or even be aware that such information is waiting until they choose to view it.

2 Context

The *Blackboard* managed learning environment divides content into a number of sections, storing course documents such as lecture slides, assignment and coursework information, general module details, dicussion groups, and many other educational resources.
In this paper we concentrate on the *Announcements* section where short announcements are posted. An example announcement is:

New Assignment 2 Deadline – Assignment 2 is now due in on the 27th of April 2004, at 4pm.

Announcements are frequently used to communicate urgent material such as lecture timetable changes, as well as less urgent material Announcements are typically very short, making them far from ideal for text classification which relies on counting the frequencies of words in text.
A prototype system is being developed that will be able to send SMS messages to students when new announcements are added to Blackboard sites for modules they are taking. We assume that it is not sensible to send an SMS message for every announcement posted to Blackboard because; (i) of the cost involved, approximately 5.5p per message per student, and (ii) because students may object if they are sent a continuous stream of largely non-urgent messages.
The Blackboard system is easily extensible. Extensions to the system can be written as Java Server Pages (JSPs), or Java Servlets, and easily integrated into a Blackboard installation. These extensions can add new functionality, interface between Blackboard and external software systems, and perform other functions.
Should our initial prototype system be a success, we intend to implement a full Blackboard extension that can send SMS text messages to students when "important" changes are made to the content of relevant Blackboard sites. Our initial prototype is not implemented as an extension because we have not yet obtained "buy-in" from the relevant technical and management personnel to modify the university's Blackboard installation. Our current system uses HTML pages and JavaScript to poll the

announcement pages of modules. The page source is then submitted to a CGI application written in Perl that can then serve the announcement text upon request to a 'bot also written in Perl. It is this 'bot that will send SMS messages. The implementation of this 'bot has not yet progressed to the point where it actually sends SMS messages, but rather this is simulated through a GUI.

The default method that our system will use is the use of special keywords. An instructor entering an urgent message into Blackboard adds the keyword *"[sms]"* into the announcement. This will be noticed by the 'bot, and a SMS message sent to students on that module. This paper describes experiments investigating whether automatic text classification can be used to remove the necessity of labelling announcements. The effort required to label texts is small. However, we believe that it is worthwhile investigating machine learning approaches for the following two reasons. First, a fully automated approach would allow all modules to be connected to the SMS messaging 'bot whether or not the instructors are aware of or willing to use the *"[sms]"* syntax. Even if an instructor is using the syntax, it is possible that they may forget to mark up urgent announcements. It would be useful if an automatic system could identify these and send a message anyway.

Messages are classified as either *low priority*, where no SMS message need be sent, and *high priority*, where a message should be sent. In the remainder of this paper we view the task of classifying announcements as that of identifying which messages are *low priority* such that no message need be sent. Because Receiver Operating Characteristic Curves are used to visualise results, we arbitrarily abstract this classification task as that of deciding that a message is of low priority and no message need be sent. In this abstraction a *true accept* is a true low priority message identified as such, a *true reject* a high priority message correctly identified. Two types of error can occur; a *false accept* is a high priority message identified as low priority, and a *false reject* is a low priority message identified as high priority. The two types of errors will not be equally deleterious. A false reject incurs expense and inconvenience to the student, while a false accept may result in students missing vital information.

For a fully automated system, we expect that the text classification procedure should be biased towards reducing the number of false accepts at the cost of increasing false rejects. Hence we should assume that messages are high priority, and send the message unless there is strong evidence that the message truly is of low priority. If the text classifier is to be used to catch accidentally unlabelled high priority messages, then the the 'bot should only send a message in the case where an announcement is unlabelled, but there is strong evidence that the message is of high priority.

Over and above the application to MLEs and SMS messages, another motivation for this research is to measure the limit of "background knowledge free" artificial intelligence techniques. The text classifier used does not use any information about announcements and teaching other than the content of a set of announcements. A skilled human attempting to classify announcements would bring a large amount of both common sense knowledge and domain (teaching and education) knowledge to bear on the task. Comparing and contrasting human and machine performance on tasks such as that described in this paper will hopefully throw light on the importance of such knowledge in short text classification.

3 Experimental methods

In this paper we detail experiments that will investigate whether it is possible to use Machine Learning based text classification techniques to identify high and low priority announcements on Blackboard.
The classifier used in our experiments is a standard naive Bayes' classifier (NBC). The use and derivation of the NBC is described in [5]. The implementation of the NBC used in this research is currently being used extensively in research on authorship attribution [3]. The NBC was selected for experiment as unpublished research into text suggests that the NBC performs better than other popular text classifiers on short texts.
If we assume that w_i is the i^{th} word from an announcement, and $p(w_i \mid P)$ is the probability of w_i appearing in an announcement of priority P, then we assign a priority to an announcement using (1).

$$P = \arg\max_{P \in \{low, high\}} \prod_{i=1}^{n} p(w_i \mid P) \qquad (1)$$

Note that (1) makes both the naive Bayes assumption of independence between evidence, and also assumes that low and high priority messages have equal prior probability. This latter assumption is true in our training sets which include equal numbers of low and high priority announcements.
As of yet we have not established the acceptable error rates for potential users of the system. As a substitute benchmark for evaluating the accuracy of any such system, we have evaluated the accuracy of human classification of messages. Both of the first two authors (RC and MB) created two files of announcements each. The files all contained 10 high priority announcements and 10 low priority announcements, making a total of 80 announcements. These were a combination of real announcements taken from our own Blackboard sites, and fictitious announcements made up for this experiment.
The announcement files were anonymised by stripping (using a program) the labelling of high and low priority, and randomly reordering the messages. Each of RC and MB then classified each other's messages. Note that there was no “training” data, and hence we were using our knowledge of education and common sense to classify these messages. We were aware that each file had ten high priority messages and ten low priority messages, making classification slightly easier than if we were not aware of this. The reason for measuring human accuracy on this task is because we expect that most instructors would be prepared to accept a human as being sufficiently accurate to judge whether SMS messages should be sent.
The first automated experiment was to use a NBC on all 80 announcements, using 10-fold cross-validation. Experiments were also performed to measure the accuracy of the classifier on announcement training sets of 20, 30, 40, 50, 60, 70, and 80 announcements. A 20 announcement training set can be constructed by simply choosing 10 random announcements each for the two authors. Note that as we only

have 80 announcements in total, the 100 randomly sampled 80 announcement training sets all had the same 80 announcements. The results for different training sets still differed due to selection of announcements in the 10-fold cross-validation. These experiments were intended to show whether prediction is more accurate when text classifiers are trained using a single instructor's announcements, and also how fast the accuracy of the classifier improves as the amount of training data increases. The latter results should allow a prediction as to whether performance of the text classification system would improve significantly given additional training data.

We have previously discussed biasing the classifier to change the balance between the two types of error. When using (1) an announcement is classified as low priority when the product of $p(w_i \mid low\ priority)$ is greater than the product of $p(w_i \mid high\ priority)$ and vice versa. This is equivalent to classifying an announcement as low priority when the ratio in (2) is greater than 1, and classifying as high priority when the ratio is lower than 1.

$$ratio = \frac{\prod_{i=1}^{n} p(w_i \mid low)}{\prod_{i=1}^{n} p(w_i \mid high)} \qquad (2)$$

We can raise the strength of evidence required to classify an announcement as low priority (and hence not send an SMS message) by requiring this ratio to be greater than a number other than 1.0. If we raise the required threshold to 1.05, then we raise the strength of evidence required to classify a message as low priority. If we lower the ratio to 0.9, then we reduce the amount of evidence required to classify a message as low priority. In this latter case it is possible that we would classify an announcement as low priority even though we have greater evidence that the announcement is high priority. As this threshold is adjusted up and down the balance between the false acceptance rate (FAR) and the true acceptance rate (TAR) will change. Note that we could also have adjusted the amount of evidence required to conclude that an announcement is low priority by defining a risk function and minimising risk rather than maximising probability, or by assigning unequal priors to the two priorities. This method of adjusting the standard of evidence required for classification has been previously used in experiments on authorship attribution, but have not yet been submitted for publication [2].

For each threshold value, the TAR and the FAR can be measured. By allowing the threshold to vary across all possible values, a number of (FAR,TAR) pairs can be gathered. Some examples can be seen in Table 1. These (FAR,TAR) pairs are plotted to create a Receiver Operating Characteristic curve [8] summarising the accuracy of the system for all values of this threshold. Any particular value for this threshold will give us TAR and FAR values for that threshold. We plot a ROC curve by mapping the FAR values to the x-axis, and the TAR values to the y-axis.

4 Results

In the human classification experiments, we achieved 0.8 (80%) accuracy. There were 16 misclassified announcements out of the total 80, with there being exactly 8 false accepts, and 8 false rejects. This gave as a FAR of 0.2, a FRR of 0.2, and hence an equal error rate (EER) of 0.2.
When the NBC was used for the full set 80 announcements , the accuracy was 0.65. The EER was 0.275. The accuracy of prediction for the files written by one author were 0.70 for announcements written by RC, and 0.625 for announcements written by MB. This compares with an average of 0.64 for training sets of 40 announcements selected from those written by either RC or MB. Figure 1 shows the results of experiments for training sets of different sizes.

Table 1.Thresholds and Error Rates

Threshold	FAR	FRR
1.028	0.050	0.800
0.991	0.275	0.275
0.945	0.800	0.050

There were 1226 distinct words in total in the 80 sample announcements. The words most indicative of each class of announcement were extracted according to the probability ratios in (3).

$$ratio = \max\left(\frac{p(word|high)}{p(word|low)}, \frac{p(word|low)}{p(word|high)}\right) \quad (3)$$

The top 30 such words are shown in Table 2.

Table 2. Most Indicative Words

Word	Indicates	Ratio
deadline	High	40.3148
3	High	36.2832
two	High	32.2518
presentation	High	28.2204
file	Low	27.7831
hand	High	24.1888
timetable	High	20.1574
monday	High	20.1574
discussion	Low	19.845
take	Low	19.845

Note the performance decreasing for the full 80 instances. This was unexpected as, all else being equal, more training data usually results in higher accuracy. Investigating the raw data shows that of the 100 experiments, there were 56 results with 0.65 accuracy, and 24 results with 0.687.

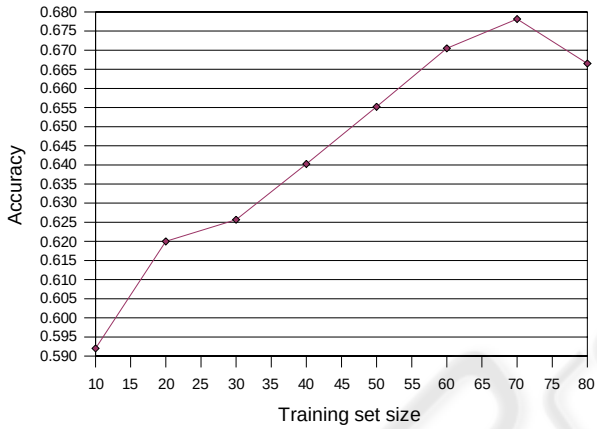

Fig. 1. Classification Accuracy Versus Training Set Sizes.

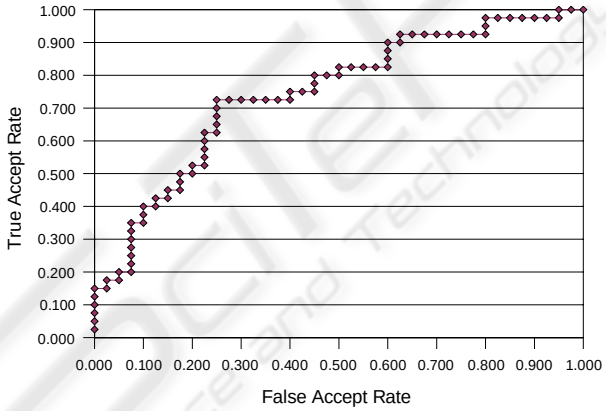

Fig. 2. Receiver Operating Characteristic Curve for Classification of All Announcements.

Figure 2 shows the Receiver Operating Characteristic Curve obtained when the cut-off for the probability ratio for accepting an announcement as low priority is allowed to range across all possible values.

Finally we extract some useful information from the raw data used in Figure 2. Arbitrarily choosing 0.05 as an “acceptable error rate”, we look for the cut-off values that give us an 0.05 FAR, and a 0.05 false reject rate (FRR), and the equal error rate (EER).

5 Conclusions

The accuracy of automated classification is currently insufficient for a fully automated system with no input from the tutor as to which announcements are sent to students. This is not surprising given the short length of most announcements. In the experiments showing how performance improved with increasing amounts of training data the performance appears to still be steadily improving, although the accuracy for the single data point for 80 announcements does confuse this issue. Also, it is not unusual for automated systems to perform worse than humans and yet prove to be of use. This can be due to cost implications, or if there is a problem getting tutors to manually label announcements. Language translation systems are an example of a technology where although human translators produce results of much higher quality, they still have many uses. Either as a "first pass" later improved by human translators, or in situations where a human translator would be too expensive and slow, such as when browsing foreign language documents on the internet.

However, it is important to note that the human classification was also quite low. Since the two authors who created the training data had taught together on a number of modules over some years, higher accuracy near to 100% might have been expected. The fact that only 80% accuracy was achieved suggests that the content of the announcements is insufficient for very high accuracies, no matter how much intelligence and background knowledge is bought to the task.

It was noted that many of the most indicative words had meaning in the context of the module, rather then being generally applicable across many modules. This is a serious problem, as if the system needs to be trained on a module by module basis, even the 80 announcements used here then many announcements will be required before the system starts working. This is a strong indication that background-knowledge free text classification will not be applicable in this domain.

Hence both human classification, applying full human intelligence and background knowledge, and a good machine classification technique indicate that announcement classification is unlikely to be useful. Our conclusions are that there does not appear to be enough information in announcements themselves to classify correctly, and larger amounts of context knowledge will be necessary.

Despite these negative results, several enhancements to the classifier are planned, most of which are fed by parallel research into author attribution. In particular, information fusion [4] approaches have shown promise in improving the confidence we can have in automated authorship attributions, if not the total accuracy. Like much technology, we would expect steady improvements in the performance of the automated system, while human requirements and performance are likely to remain static. We are encouraged by the comments of Christensen *et al* (2001) who argue that technology advances faster than user requirements. However, our results from human classification do call into question whether any amount of technology will really be able to solve this problem.

Whether sending a SMS text that should not be sent, or failing to send an SMS text that should not be sent is a greater error is a question for management, not technology. The ability to tweak the classifier to achieve different balances between false accept and false reject means that different management policies can be implemented in the system. This is also important given that tutors may (or may not) choose to override

the system with *[sms]* and *[nosms]* messages in different ways, and may prefer the ability to customise automatic text sending according to their own preferences.

At present we are using a generic text classification method and implementation. It may be possible to improve performance by building a recogniser that extracts features from the text particularly relevant to the classification being performed. For example noting text patterns such as *"URGENT MESSAGE"* and *"must immediately"*, *"no rush"*, atypical use of all capitals, and other features might be useful in improving classification.

Announcements on *Blackboard* and other MLEs are not the only web-based applications where very small sections of text need to be classified. Therefore we feel that work on short text classification, as well as the management and usability issues concerning text classification of limited accuracy, will have wide application in many online contexts.

REFERENCES

1. Beasley, N., & Smyth, K. 2004. Expected and actual student use of an online learning environment: A critical analysis. *Electronic Journal on E-Learning* **1:** 43-50.
2. Clement R. (unpublished). Verifying authorship. Unsubmitted draft paper.
3. Clement R., & Sharp, D. 2003. Ngram and Bayesian classification of documents for topic and authorship. *Literary and Linguistic Computing* **18** (4): 423-47.
4. Kuncheva, L. 2004. Combining Pattern Classifiers: Methods and Algorithms. Wiley.
5. Mitchell, T. 1997. *Machine Learning.* McGraw-Hill.
6. Richter, U. 2001. An exploration of using web enhanced teaching and online communication in a conventional context. *Proceedings of the Fifth Romanian Internet Learning Workshop.* Romania. http://rilw.emp.paed.uni-muenchen.de/2001/papers/richter.html
7. Stiles, M. 2000. Effective learning and the virtual learning environment. *Proceedings: EUNIS 2000 - Towards Virtual Universities.* Poland. http://www.staffs.ac.uk/COSE/cose10/posnan.html
8. Swets, J. 1988. Measuring the accuracy of diagnostic systems. *Science* **240:** 1 285-93.
9. Ubon, A., & Kimble, C. 2004. Exploring social presence in asynchronous text-based online learning communities (OLCs). *Proceedings of the 5th International Conference on Information Communication Technologies in Education* 292-7. Greece.