The FAQ Robot – Managing Large-class E-mail Correspondence

Frode Eika SANDNES
Faculty of Engineering, Oslo University College, P.O. Box 4, St. Olavs plass, N-0130 Oslo, Norway, E-mail: frodes@iu.hio.no, URL: http://www.iu.hio.no/~frodes/

Baard H. Rehn JOHANSEN
Faculty of Engineering, Oslo University College, P.O. Box 4, St. Olavs plass, N-0130 Oslo, Norway, E-mail: baard@rehn.no URL: http://baard.rehn.no/

Nils Einar EIDE
Faculty of Engineering, Oslo University College, P.O. Box 4, St. Olavs plass, N-0130 Oslo, Norway, E-mail: nils_einar.eide@chello.no

Andreas N. BLAAFLADT
Faculty of Engineering, Oslo University College, P.O. Box 4, St. Olavs plass, N-0130 Oslo, Norway , E-mail: andreas@blaafladt.no URL: http://andreas.blaafladt.no

KEYWORDS: *student queries, large class teaching, asynchronous communication, document classification and clustering, frequently asked questions*

ABSTRACT: *In classes with a large number of students the number of e-mail requests from students can be quite large at times. For example, when a new project has been published students often have questions regarding specific details and ambiguous aspects of the project specification. Often these queries can be classified into a specific set of categories, for example questions regarding the formatting of the project report or clarification of weakly stated problem parameters. In this project we have experimented with document classification technology and developed a tool for automatically classifying student questions in the form of short e-mail messages and pseudo-automatically responding to these requests. This paper describes the problems associated with e-mail correspondence when teaching large classes and how we designed the FAQ (frequently asked questions) robot to assist in reducing these problems. Further, preliminary experimental evaluations on how well the tool manages to classify and respond to incoming requests are provided. Our findings point in the direction that one cannot completely rely on a totally automatic tool for sorting and responding to incoming e-mail from students. However, document classification technology and the FAQ robot provides a useful companion that can help the teacher quickly get an overview of the class situation and more rapidly help and guide students in the right direction.*

1 E-MAILS AND THE DIFFICULTIES OF ONE-TO-ONE CORRESPONDENCE

E-mail communication between students and teachers is commonplace in higher education [HEDGES, K. & MANIA-FARNELL, B. 1999]. The asynchronous nature of e-mails is convenient to both students and teachers given the fact that both students and teachers have more demanding, multifaceted and complex lifestyles than before. Although not anonymous the use of e-mail as a means of communication is less intimidating to some students than approaching the teacher in person – lowering the threshold of making contact. Most studies suggest that increased student teacher interactions in all forms leads to a better learning environment for students. Students that seek immediate feedback on their coursework perform better than students that do not [CLARKE, M., BULTER, C, SCHMIDTH-HANSEN, P & SOMERVILLE, M. 2004]. Further, conversations and enquiries from students help the teacher better understand the difficulties faced by the student allowing the teacher to make essential adjustments during a course [CASHIN, W. E. 1988].

The increased electronic interaction comes at a cost – additional teacher workload. In situations with large classes comprising 50 or more students the teacher becomes the bottleneck having to conduct one-to-one e-mail correspondence with all the students. For very large classes, say for instance 400 students,

the one-to-one communication scenario becomes next to impossible to manage. From the teachers' angle the correspondence with student comes in addition to ordinary faculty correspondence, correspondence with external contacts, junk mail and the daily dose of spam. Simultaneously, teachers are expected to become more efficient, are given additional duties and responsibilities and handle increased teaching loads.

The student teacher correspondence follows predictable patterns. First, it can be classified into four major categories – questions regarding assignments, technical questions regarding lectures or the curriculum, exam or assessment related issues and other. Second, questions related to assignments and exams have a tendency to be concentrated around certain intervals in time. It is common for the teachers to receive many queries just before a deadline or an exam is due. Then, it is also common to get queries immediately afterwards regarding, for instance, the assessment. Other queries such as technical questions are less predictable and usually arrive at a random but on average constant rate related to the class size.

During these busy periods it can be hard to respond to all student queries adequately and in a timely manner. However, many of the queries are similar in nature. Most teachers have probably experienced the problem of answering the same question multiple times. To overcome such difficulties some teacher employ FAQ (Frequently Asked Questions) lists that they post of the course website (see for instance [HUANG, H. P. & LU, C.-H. 2003]). Once a teacher gets a novel question the teacher will add the question to the FAQ together with an answer. The teacher will instruct the students to consult the FAQ before asking any questions. In this way, the amount of correspondence is reduced as many of the answers can be found on the FAQ, and several teachers report successes in using FAQs. However, the drawback of FAQs is that they require effort, diligence and discipline to be maintained adequately and continuously. Consequently, the FAQ strategy is not appealing to everyone. It may also not suit the pedagogical teaching style of some teachers. Also, the idea of a list with all the "correct" answers is more suitable to the engineering and scientific community than it might be for the social science community where there often is no single "correct" answer.

This work is trying to address these issues. The general idea is that by the means of modern document classification techniques the teaching related correspondence effort of the teacher can be reduced. A FAQ robot is designed to automatically assess all incoming messages and catalogue these accordingly. Instead of being faced with a full and overwhelmingly long inbox of messages the teacher is faced by a set of incoming messages prearranged into categories. The teacher can then respond more efficiently to messages in bulk, by writing fewer more general replies. The pedagogical rationale behind this strategy is that students should get more immediate and timely feedback from the teacher and the quality of the feedback should be better. For example, imagine 10 students submitting similar queries on a given topic. If the teacher responds individually to all these requests the teacher will suffer from mental fatigue and the quality of the responses will be dispersed across the 10 individual responses. It will also take a long time to compose the 10 replies. Alternatively, responding to one message takes much less time and effort from the teacher and the quality of the response will therefore be better.

The difficulties of handling large number of requests are also commonplace in large organisations.. Research effort has gone into the automatic retrieval of answers from existing question answer lists based on queries, for example the VIENA classroom system, which uses lexical similarity [WINIWARTER, W., KAGAWA, O. & KAMBAYASHI, Y. 1996; WINIWARTER, W. 1996], the FAQ**F**inder system [BURKE, R., HAMMOND, K., KULYUKIN, V., LYTINEN, S., TUMURO, N. & SCHOENBERG, S. 1997; KULYUKIN, V. A., HAMMOND, K. J. & BURKE, R. D. 1997] which uses semantic knowledge, the FAQIQ system which uses case based reasoning [LENZ, M. & BURKHARD, H. D. 1997]. In a different approach [SNEIDERS, E. 2002] are able to answer questions based on information in relational databases using templates. Common to all these strategies is that they are based on already existing FAQs. Our approach makes no such assumptions.

1.1 Application: increased student-teacher interaction in lectures

The proposed FAQ robot is not limited to out-of-class scenarios, but can also be applied to a large-class lecture scenario. It is generally agreed that lectures is not an optimal learning environment. Research has shown that class size affects academic performance (see the classic study [GLASS, G. V. & SMIDTH, M. K. 1979]). However, real-life limitations often make lectures the only viable option.

Further, the students are familiar with lecture style of learning. Research by Banham and Elender [BANHAM, J. & ELENDER, F. 1995] suggests that learning can be stimulated in large classes by increasing the range of work in which students are actively involved. [FELDER, R. M. 1997] recommends getting students actively involved and help them develop a sense of community. Inspired by similar ideas [SANDNES, F. E. & TALBERG, O. 2004] proposed a strategy for increasing the student-teacher interactions in large class settings by the means of mobile phones as student terminals. In their mobile teaching system GameShow the students could anonymously, easily and freely interact with the teachers using their mobile phones. Students could send gists, respond to quizzes, polls and submit questions in real-time during a lecture. All the information was summarised on the projected screen in the front of the lecture theatre. Although, Sandnes and Talberg were unable to experiment with their system in practice, it was anticipated that a bottleneck in the system could be the teacher having to assess a vast amount of information coming from the students tapping away on their mobile handsets. By incorporating the FAQ robot into the GameShow system, the teacher will get a more organised and systematic overview of the class situation and will be able to handle a large number of requests from a very large class (for instance 400 students). The teacher will more quickly be able to adjust and coordinate the lecture according to the student feedback – a characteristic essential in successful lecturing [CASHIN, W. E. 1985]. As research by [GOLDSTEIN, G. & BENASSI, V. 1996] shows – students' perception of a good lecturer is directly related to what they perceive as a good discussion leader.

One important advantage of the GameShow system is that existing infrastructure is exploited, hence the mobile handsets. New educational technologies have a tendency to fail because institutions are unable to invest in the required infrastructure (see [DEDE, C. 1998]).

2 THE FAQ ROBOT

The FAQ robot employs techniques borrowed from web mining (see [CHAKRABARTI, S. 2002] for a general introduction to web mining) and terminology mining based on text corpora (see for example [JUSTESON, J. & KATZ, S. 1995]. The FAQ robot takes a set of messages as input and produces a set of message clusters as output, i.e. related messages are clustered together, and unrelated messages are placed in different clusters. The step comprises several phases. First, the system must retrieve the messages, then each message is pre-processed and transformed into a text vector. The set of text vectors representing the set of messages are used as input to the clustering algorithm for then to compute the most suitable clusters and finally the results are presented to the recipient (user).

Messages are submitted to the system in one of two ways, either using an online web-form or via traditional e-mail. A dedicated e-mail account is set up and the FAQ robot polls the inbox at regular intervals to check for new incoming messages.

The inbound messages are processed as follows: Each message is treated as a separate entity and used as a basis for computing a word vector. A word vector represents a set of words as a vector, where each word in a dictionary is assigned a specific position in the vector, thus the size of the vector equals the size of the dictionary used. The presence of a word is marked by a positive non-zero integer, where the value represents the number of times the word occurs in the text. A zero value denotes the absence of a specific word. Cleary, different messages will usually have different word vectors in the word space.

To generate a word vector the text is organised into individual words. The first step is to filter high frequency words, also known as *stop words* [WILBUR, J. W. & SIROTKIN, K. 1992; HO, T. K. 1999]. This is achieved using a stop word dictionary computed from word frequency lists [KILGARRIFF, A. 1997]. Next, word stemming is employed to obtain the general form of words [PORTER, M. F. 1980] with the purpose to reduce the size of the word vectors. Then, a dictionary-based spell checking technique is used. I.e. a reference dictionary comprising of all the possible valid words in the language in all grammatical forms is used. If an entry in the text dictionary cannot be found in the reference wordlist then the entry is tagged as a potential incorrect spelling. All entries marked as potential incorrectly spelled words are crosschecked against the reference wordlist using Metaphone [PHILLIPS, L. 1990]. Metaphone is a technique, inspired by SOUNDEX for matching words based on their phonetic sound and it is particularly suitable for spell checking applications. Entries with no Metaphone match in the dictionary are considered special terms. Entries with a metaphone match are most likely incorrectly spelled words (see [KUKICH, K. 1992] for an excellent survey of automatic spelling correction techniques). Then, each

instance of the remaining words is counted and represented in the word vector. The word vector therefore represents the potentially interesting words that are characteristic of the question [KILGARRIFF, A. 1996].

The word vectors are clustered using the K-means algorithm – a classic and widely known and effective algorithm (see [DUNHAM, M. H. 2003]. In clustering the words are represented as vectors in the word space, and the purpose of the clustering algorithm is to assign word vectors that are similar to the same cluster in the vector space and assign word vectors that are different to different clusters. Two vectors are similar if the distance between the two vectors is small, and two vectors are dissimilar if their distance is large. One popular distance measure is the Eucludean distance. The K-means algorithm works as follows: a set of vectors is to be clustered into K clusters. Initially, the vectors are assigned arbitrarily to the K clusters. Then the mean vector for each cluster is computed. Then, the vectors are reassigned to the cluster to which they are closest and the cluster means are recomputed. The process is repeated until some convergence criteria are met.

Finally, the results of the clustering algorithm are presented to the user as a pre-catalogued set of messages.

2.1 Example: clustering computer science student queries

Imagine there are two weeks left before an exam in the course "C++ programming" and the teacher responsible for the course receives the following six questions via e-mail from five different computer science students following the course:

1: "When will the examm results be out?"
2: "In exam question 4.5 what do you mean by operator overloading?"
3: "Last years exam was too difficult, will we get questions on operators?"
4: "Can I get an extension on the assignment deadline?"
5: "Will there be a lecture on Thursday?"
6: "Could you please go over previous exam papers on Thursday?"

Further, imagine that the FAQ robot uses the following list of common high-frequency stop words:

Stop words: *an, be, by, can, could, difficult, do, get, go, I, in, last, on, out, over, please, previous, the, too, was, we, what, when, will, you, there*

Based on the corpus comprising of the five questions we have the following dictionary or word vector space. Note that the numbers preceding the words indicate their position in the word vector.

Dictionary: *0: assignment, 1: deadline, 2: exam, 3: extension, 4: lecture, 5: mean, 6: operator, , 7: overloading, 8: paper 9: question, 10: result, 11: Thursday, 12: year*

The question *"When will the examm results be out?"* is converted to a word vector as follows. First, all the stop words are removed, namely "when", "will", "the", "be" and "out" and the two words "examm" and "results" remain. The word "examm" is not in the dictionary and by using the Metaphone algorithm we find that the word is very similar to "exam". The word is therefore assumed to be a misspelled instance of the word "exam". Further, the stemming algorithm removes the suffix s from the word "results" yielding the word "result". The two words "exam" and "result" are in position 2 and 10 in the dictionary respectively and phrase 1 therefore yields the following word vector.

1: [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0]

The same process is repeated for the remaining five questions resulting in the word vectors:

2: [0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0]
3: [0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1]

4: [1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
5: [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0]
6: [0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0]

Then, the distances between the different word vectors are computed (see Table 1):

Table 1 – Distance between a set of six word vectors

	1	2	3	4	5	6
1	0	5	3	5	2	1
2		0	3	9	6	6
3			0	7	6	5
4				0	5	6
5					0	3
6						0

According to this distance measure question 1 is similar to questions 6, 5 and 3 in that order. Question 2 is only similar to question 3 and question 5 is similar to question 6. Question 4 is very dissimilar to all the other questions. Consequently, based on the K-means algorithm with K=4 we could group the questions as follows with the following keyword summaries:

Exam, results;
1: "When will the examm results be out?"

[Manual reply: probably three weeks after the exam.]

Exam, question, operator:
2: "In exam question 4.5 what do you mean by operator overloading?"
3: "Last years exam was too difficult, will we get questions on operators?"

[Manual reply: I have removed operator overloading from this years syllabus.]

Extension, assignment, deadline:
4: "Can I get an extension on the assignment deadline?"

[Manual reply: I'm sorry. Not possible.]

Thursday:
5: "Will there be a lecture on Thursday?"
6: "Could you please go over previous exam papers on Thursday?"

[Manual reply: Next Thursday I will review past exam papers.]

The result is four categories based on the six original questions. In this example the six questions can easily be answered with four answers.

2.2 The FAQ robot inner workings - implementation

The implementation consists of the following components: *Controller-module***:** Controls the flow of information through the system (see Figure 1). The controller modules can be interconnected in an arbitrary manner, i.e. the output of one module is routed to the input of another module etc. Messages are processed as they travel through the various modules on their path to their destination. A typical message-path comprises an *input-module*, a set of *pre-parsers*, a *parser-module*, a set of *filters*, a *sorter-module* and an *output-module*. Multiple instances of a module can be created and used in different part of the message chain.

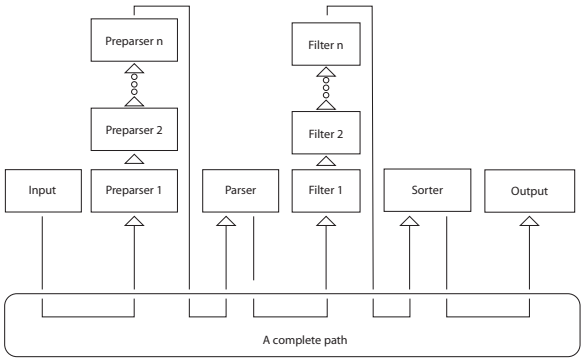

Figure 1 - Information flow

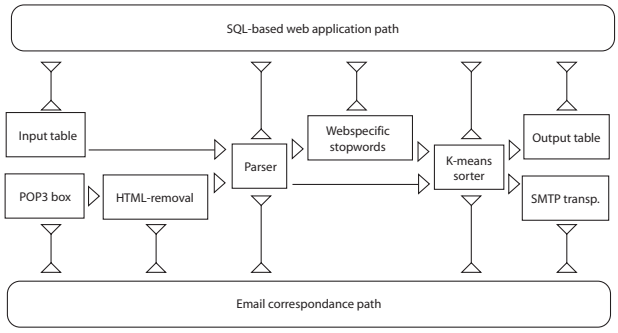

Figure 2 - System architecture

The controller is equipped with an RMI (Remote Method Invocation) interface [SUN 1999]. This feature allows systems with a high degree of interactivity to be configured. I.e. the system can be configured to provide immediate feedback to instant inbound messages. A special message-path configuration is required, which does not contain conventional input and output modules. This implementation also allows modules to be interconnected to form a tree-structure where all non-leaf nodes are routers that forward messages to other modules using the RMI interface. For instance, a controller can be configured to identify the language of a message and forward the message to the corresponding controller for the target language of that message. Messages can be rejected and re-routed to different branches of the graph if problems are detected by modules higher up in the tree.

Input-modules retrieve and convert external data into the internal representation used by the system. The current set of input modules can read POP3 (Post Office Protocol) and IMAP (Internet Mail Access Protocol) mailboxes, and external JDBC(Java DataBase Connectivity)-compliant database tables provided the correct fields are appropriately configured.

Preparsers modify the incoming messages before they are tokenised into sentences and words. Removal of HTML-tags is a common pre-parser filter operation.

Parsers tokenise texts into sentences and words. A message must be parsed before further processing can take place.

Filters alter or remove words or phrases from messages. For example, a spellchecker replaces incorrectly spelled words with their corresponding correctly spelled words, while a stop-word filter removes words that are not relevant to content of the message.

Sorter-modules attach answers to incoming messages, or signals to the controller that there are no matching answers to the incoming question.

Output-modules return messages back to their source provided they belong to the same category. For example, POP3 and IMAP input messages are returned via the SMTP (Simple Mail Transfer Protocol) output module. Further, the database input messages can be returned to another external JDBC-compliant database using references created by the input module.

GUI (Graphical User Interface)-controllers are equipped with swing-based GUI components which allows for direct user-controller-interaction. The current GUI controller can be connected to multiple controllers simultaneously. A HTML-based web interface is not currently provided, but is planned for a future release. The application is written in Java, currently using MySQL 3.23 for persistent storage. The application is developed and tested using Sun's JRE 1.4.2 (Java Runtime Environment).

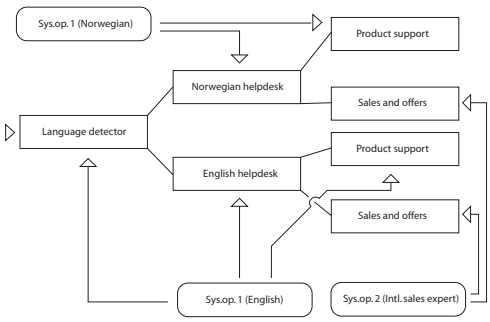

Figure 3 - Using multiple controllers

3 PRELIMINARY RESULTS AND DISCUSSION

The project is still young and our test suites are very limited. There are several difficulties associated with acquiring good test data. Firstly, lecturers are often quick to remove e-mail as e-mails from students quickly accumulate into unmanageably large collections. Lecturers that do store their e-mail correspondence with students are often reluctant to pass on what they view as their private correspondence to "strangers". Further, the data protection act limits certain aspects of message acquisition and storage. Finally, once test messages are acquired they need to be manually classified and tagged – a job that is both laborious and error-prone.

We currently possess four message collections. Two of the collections comprise student-to-teacher correspondence in the courses "web-applications" and "mobile computing" given at the Faculty of Engineering, Oslo University College autumn and spring of 2003 respectively. These are all limited by the fact that about 90% of the messages in each set can hardly be describes as similar. The third message collection consists of questions from a Norwegian textbook on web application development [SANDNES, F. E. 2002]. This set of questions is manually classified into parts and chapters. Unfortunately, the "quick fix" chapter categorisation is inaccurate and superficial as there are relationships between questions from different chapters in addition to the fact that there are not necessarily any relationship between questions contained within the same chapter. Understandingly, the questions of a textbook are typically designed to cover as wide ground as possible instead of repeating the same aspect of the content.

Because of these difficulties a forth set of 80 questions were manually created artificially, based on 15 unique themes. Questions associated with one theme were formulated differently, but asking the same basic question. The purpose was to simulate the typical situation a teacher experiences when unclear instructions are given. For example: "When do we hand in the assignment?"`, "When is the assignment due?", "What is the assignment deadline?", "What's the deadline?" etc.

Figures 4 and 5 illustrate how the FAQ-robot performs on the forth test-suite. Figure 4 illustrates the convergence of the FAQ robot as the number of messages increases. Each sample represents the mean taken over an interval of 10 messages. Note that the messages of the test-suite are shuffled into a pseudo-random order before presented to the FAQ-robot. The horizontal axis indicates the number of messages received accumulatively, and the vertical axis represents the probability of creating a new cluster. Initially, when the system is presented with 0 to 10 messages the probability of generating a new cluster is one. Once the number of messages is above 10 the probability of creating a new cluster rapidly decreases to approximately 0.3. After about 30 messages are received the probability of generating a new cluster has stabilised at around 0.2, and this number is gently reduced to 0.1 as the 80-message-mark is reached.

Figure 5 illustrates the success rate when the FAQ-robot is applied to the forth test-suite. The horizontal axis represents the number of messages processed accumulatively, and the vertical axis represents the ratio of successfully classified messages. A successful classification is a classification where the FAQ-robot classification matches the manual classification, i.e. all the questions in the test-suite are manually classified and labelled with its respective class-id prior to the experiment. After the clustering each cluster is labelled with the classified according to the majority of members of a given classification. For example, if six messages belonging to class A is assigned to one cluster, and four messages from class B are assigned to the same cluster, then cluster A will be the "correct" cluster and the four B-class messages will be incorrect members of the cluster. Each sample represents the mean taken over an interval of 10 messages.

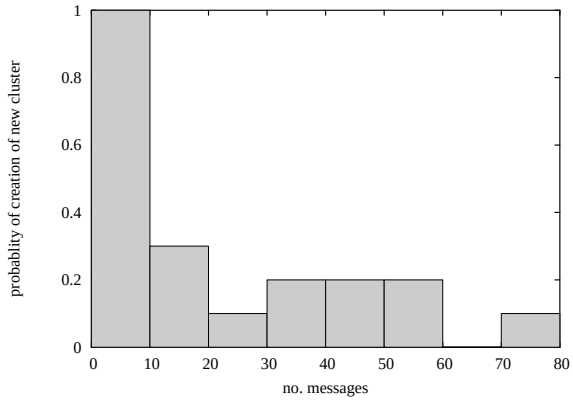

Figure 4 - Clustering probability

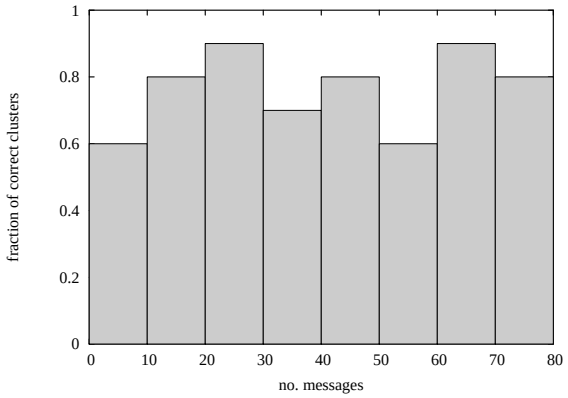

Figure 5 - Success rate

Figure 5 shows that initially about 60% of the messages are correctly clustered and classified and towards the end about 80% of the messages are correctly clustered and classified. This is result very good. However, this is as expected since the results are based on artificial data.

We are currently gathering more data for further testing, and some more realistic results will be posted on the project website (http://www.digimimir.org).

4 CONCLUSIONS

In this paper the FAQ robot was presented. The FAQ robot is a message management tool that can be used in an offline out-of-class setting as well as a lecture setting to simplify the correspondence with students. The intentional pedagogical benefits of the FAQ robot are shorter teacher-to-student response times and higher quality responses. It is an established fact in pedagogy that timely feedback is an essential part of effective learning. Further, the workload of the teacher can be reduced and the tool may help the teacher more quickly identify and understand the class situation and respond to potential difficulties and weaknesses. Further information about the FAQ robot and a prototype open source implementation can be downloaded from http://www.digimimir.org

ACKNOWLEDGEMENTS

The authors are grateful to Sigmund Straumsnes of the Engineering faculty at Oslo University College for providing e-mail test data.

REFERENCES

BANKHAM, J. & ELENDER, F., ”Applying person-centered principles to teaching large classes”, British Journal of Guidance and Counselling 23:2, 1995.

BURKE, R., HAMMOND, K., KULYUKIN, V., LYTINEN, S., TUMURO, N. & SCHOENBERG, S. ‘Natural Language Processing in the FAQ Finder System: Results and Prospects’, in Working Notes from AAAI Spring Symposium on NLP on the WWW, 1997, 17-26.

CASHIN, W. E. ”Improving Lectures”, IDEA paper 14, Kansas State University, 1985.

CASHIN, W. E. “Student Ratings of Teaching: A Summary of the Research”, IDEA paper 20, Kansas State University, 1988.

CHAKRABARTI, S. ‘Mining the Web: Analysis of Hypertext and Semi Structured Data’ Morgan Kaufmann**,** 2002.

CLARKE, M., BULTER, C, SCHMIDTH-HANSEN, P & SOMERVILLE, M. ‘Quality assurance for distance learning: a case study at Brunel University’, British Journal of Educational Technology 35:1, 2004, 5-13.

FELDER, R. M., ‘Beating the numbers game: effective teaching in large classes’, Proceedings of ASEE Annual Conference, Milwaukee, June, 1997.

DEDE, C. ‘Six challenges for educational technology’, Learning with technology (ed. Dede, C.) ASCD Yearbook, 1998.

DUNHAM, M. H. ‘DATA MINING: Introductory and Advanced Topics’, Prentice Hall, 2002, 140-142.

GLASS, G. V. & SMIDTH, M. K. ‘Meta-analysis of research on class size and achievement’, Educational Evaluation and Policy Analysis 1, 1979, 2-16.

GOLDSTEIN, G. & BENASSI, V. ‘Students' Perceptions of Excellent Lecturers and Discussion Leaders’, Journal on Excellence in College Teaching 7:2, 1996, 81-97.
HEDGES, K. & MANIA-FARNELL, B. ‘Using E-mail to Improve Communication in the Introductory Science Classroom’, Journal of College Science Teaching 28:3, 1999, 198-203.
HUANG, H. P. & LU, C.-H. ‘Java-Based Distance Learning Environment for Electronic Instruments’, IEEE Transactions on Education 46:1, 2003, 88-95.
HO, T. K. ’Fast Identification of Stop Words for Font Learning and Keyword Spotting’, Proceedings of the 5th Int'l Conference on Document Analysis and Recognition, 1999, 333—336.
JUSTESON, J. & KATZ, S. ‘Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text’, in Natural Language Engineering 1:1, 1995, 9-27.
KILGARRIFF, A. ‘Which words are particularly characteristic of a text? A survey of statistical approaches’, Proceedings of AISB Workshop on Language Engineering for Document Analysis and Recognition, Sussex University, 1996, 33–40.
KILGARRIFF, A. ‘Putting frequencies in the dictionary’, International Journal of Lexicography 10:2, 1997, 135-155.
KUKICH, K. ‘Techniques for Automatically Correcting Words in Text’, ACM Computing Surveys 24:4, 1992, 377-439.
KULYUKIN, V. A., HAMMOND, K. J. & BURKE, R. D. ‘An Interactive and Collaborative Approach To Answering Questions for an Organization’, Technical report TR-97-14", Dept. Computer Science, University of Chicago, 1997.
LENZ, M. & BURKHARD, H. D. ‘CBR for Document Retrieval - The FAllQ Project’, Case-Based Reasoning Research and Development (ICCBR-97), Lecture Notes in Artificial Intelligence No. 1266, Springer-Verlag, Berlin, 1997, 84-93.
PHILLIPS, L. ‘Hanging on the Metaphone’ *Computer Language*, 7:12, 1990, 39-43.
PORTER, M. F. ’An algorithm for suffix stripping’ Program 14:3, 1980,130--137.
SANDNES, F. E. ’Moderne Applikasjonsutvikling i Java for Web: tynne klienter og fete tjenere’ Tapir Akademisk Forlag, 2002.
SANDNES, F. E. & TALBERG, O. ‘Mobile Phones in the Lecture Theatre - Using Wireless Technology as a Pedagogical Aid’, in ENGINEERING EDUCATION AND RESEARCH - 2004: A Chronicle of Worldwide Innovations (ed Win Aung),. Begell House, 2004. (to appear)
SNEIDERS, E. ‘Automated Questioning Answering: Template Based Approach’, PhD Thesis, Dept. Computer and System Science, Stockholm University, 2002.
SUN ’JavaTM Remote Method Invocation Specification’, Sun Microsystems, Revision 1.7, 1999.
WILBUR, J. W. & SIROTKIN, K. ’The automatic identification of stop words’ Journal of the American Society for Information Science 18, 1992, 45-55.
WINIWARTER, W., KAGAWA, O. & KAMBAYASHI, Y. ‘Applying Language Engineering Techniques to the Question Support Facilities in VIENA Classroom’, Database and Expert Systems Applications, 1996, 613-622.
WINIWARTER, W. ‘Collaborative hypermedia education with the VIENA Classroom system’ In Proc. 1st Australasian Conf. on Computer Science Education ACSE'96), 1996, 337—343.