Annotating educational discussion boards to help students who are blind

Maria Luisa Sapino*

Dipartimento di Informatica,
Università di Torino, Torino, 10149, Italy
E-mail: mlsapino@di.unito.it
*Corresponding author

K. Selçuk Candan and Jong Wook Kim

CSE Department, Arizona State University,
Tempe, AZ 85287, USA
E-mail: candan@asu.edu E-mail: jong@asu.edu

Fabrizio Antonelli

Dipartimento di Informatica,
Università di Torino, Torino, 10149, Italy
E-mail: fabrizio.antonelli@alice.it

Abstract: Educational discussion boards are growing in use as they help students share knowledge and doubts in a working/studying environment. Although these interactive tools are highly effective in learning environments, their usability by blind students is very poor. In this paper we develop techniques to improve accessibility of educational content for students who are blind. Threads of messages in discussion boards evolve with new postings, thus just by investigating the subject headings or contents of earlier postings in a message thread, students may not be able to guess the contents of the postings deeper in the hierarchy. In order to overcome the navigation obstacle for users, it is essential to develop techniques that help identify how the content of a discussion board grows. We develop a technique to organise messages in a message board, by automatically classifying and annotating postings to guide users through discussion segments relevant to their navigational goals.

Keywords: website accessibility; navigation in message boards; rule-based classification.

Reference to this paper should be made as follows: Sapino, M.L., Candan, K.S., Kim, J.W. and Antonelli, F. (2007) 'Annotating educational discussion boards to help students who are blind', *Int. J. Continuing Engineering Education and Life-Long Learning*, Vol. 17, Nos. 4/5, pp.294–318.

Biographical notes: Maria Luisa Sapino got her MS and PhD Degrees in Computer Science at the University of Torino, where she is currently an Associate Professor. Her initial contributions to computer science were in the area of logic programming and artificial intelligence, specifically in the semantics of negation, and in the abductive extensions of logic programs.

Since mid-1990s she has been applying these techniques to the challenges associated with heterogeneous and multimedia data management. She developed novel techniques for similarity based information retrieval, and she is currently working on web accessibility for users who are visually impaired, and on temporal aspects in distributed multimedia presentations in the presence of resource constraints.

K. Selçuk Candan is an Associate Professor at the Department of Computer Science and Engineering at the Arizona State University. He received his PhD in 1997 from the Computer Science Department at the University of Maryland. He has worked extensively on the integration of heterogeneous multimedia information. His current research include data processing and indexing for sensory data, distributed data management for efficient dynamic content delivery, and adaptive information technologies for helping individuals who are blind. He has published over 80 papers in respected venues. Most recently, he is an editorial board member of the *Very Large Databases* (*VLDB*) journal and the *ACM SIGMOD Digital Symposium Collection* (*DiSC*).

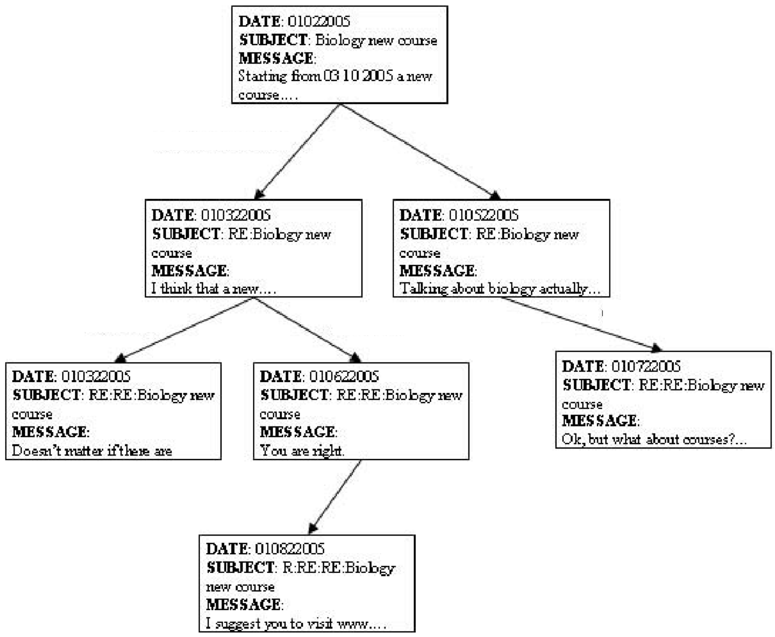
Jong Wook Kim is a PhD student at the Department of Computer Science and Engineering, Arizona State University, AZ, USA. He received his BS Degree from Korea University in 1998 and his MS Degree from KAIST in 2000. Before joining ASU, he conducted research at Digital Media Research Lab in LG Electronics. His primary research interests are in the areas of web data mining, information retrieval, and database systems. In particular, his current research concentrates on indexing and mining for non-atomic web data, such as a navigation hierarchy and a discussion boards, and application of these techniques for development of navigational helps for individuals who are blind.

Fabrizio Antonelli received his degree in Computer Science from the University of Torino in 2005. In 2003, he earned an internship at Fiat Avio, where he worked within a European project for knowledge management for airplane engine factories. In 2005, he visited the Department of Computer Science and Engineering at Arizona State University and contributed to the iCare project for improving accessibility of educational websites and tools. He is currently a researcher at the Telecom Labs, Italia.

1 Introduction

Many organisations (such as companies, universities, schools, e-learning societies) are developing online tools to enable their users share and exchange information. One of the most important tools in this category is the *discussion board*, where users can post messages to ask a question, to provide answers to other' questions, or to post an announcement (Figure 1). In general, it is very common for discussion boards to contain postings about several different topics at the same time (within a single thread or across multiple threads) and it is difficult for a user who is blind to locate a posting relevant to her task. Suppose a user who is blind wants to know if anybody knows the answer to a specific question or can provide suggestions about a specific topic. It would be very convenient for the user, if the system could assist by providing her with a list of postings matching her interests (based on the criteria she gave), without making her scan the entire board.

Figure 1 An example discussion board tree



With the aim of reducing the navigational load of blind students, we are developing a software interface, called *iCare-Assistant*, which provides context- and task-dependent navigational guidance when accessing online educational materials that are already available for the use of sighted students. iCare-Assistant provides a transparent interface between an existing educational system (such as Blackboard, a commercial educational software (http://www.blackboard.com)) and the blind student, and adds context and keyword-based navigation (query) facilities to locate the desired information with a few interactions only, reducing the navigational load in accessing information. State-of-the-art browser-based interfaces and existing navigational helps, such as site maps and visual cues, alleviate this load for only sighted users and are generally not applicable to dynamically growing content. Instead, we employ transparent guidance and dynamic adaptation techniques (Donderler et al., 2003, 2004) in *iCare-Assistant* to help students without sight. Such dynamic adaptation and guidance require an understanding of the inherent, yet implicit, structures of the content at the educational websites.

Text mining techniques (Cohen, 1996; Liu et al., 2003) have been successfully developed and applied in many business intelligence applications. Discussion records, however, cannot fully benefit from these techniques, because of their subtle, often implicit, dependencies upon each other. It is often the case that an individual posting carries a very limited amount of information (which makes text mining techniques of very little use on such postings), whereas a collection of interrelated postings in the discussion board forms a context, which carries more information, if properly identified. Moreover, standard text mining techniques and indexing solutions do not apply to message boards since the collections of messages are highly dynamic. The challenge, in this case, is the discovery and organisation of such information possibly coming from different messages, given the fact that:

- several different topics can be discussed in a single message, whose title/subject is not necessarily required to match the message content
- different aspects of the same topic may be found in various postings.

Contributions of this paper

In this paper, we address the problem of knowledge discovery and presentation from message-boards. More specifically, we focus on the challenges associated with accessing *educational discussion boards*, i.e., discussion boards used by teachers, assistants, and students, to exchange information. We concentrate on the problems of classifying the posted messages and discovering different relationships among messages posted on the board. Based on the resulting classification, appropriate indexing techniques can be developed to improve the information accessibility. In particular, we develop techniques to improve accessibility for students who are blind, and therefore cannot benefit from the visual help offered by well engineered websites.

The logical organisation of the messages is described by means of a labelled tree, whose nodes are associated to the postings and whose labelled edges characterise the different inter-dependency relations existing between messages. The labelling of the messages is realised by means of a rule-based system (JESS), which associates (possibly multiple) scores to the automatically extracted interdependency relations between pairs of messages. Among all possible labels for a given message, the ones with the highest score are chosen as the assigned classification for the message. One of the advantages of the use of a rule-based system is its dynamic adaptation to the board content: at any point in time, the arrival of new messages might fire previously inhibited rules and induce an appropriate revision/update of the classification for some message, on the basis of the newly added information.

The paper is organised as follows. In Section 2 we give a brief overview of the related work. In Section 3 we introduce the architecture of the navigation support module. Section 4 describes the method we use to represent the information associated to the messages, both content and metadata. The rule-based classification module is presented in Section 5. Section 6 presents usage strategies for the module, and Section 7 illustrates experimental results. Finally, Section 8 contains concluding remarks.

2 Related work

In this section, we present the related work in the domains of adaptive and assistive web and educational technologies, adaptive hypermedia, and message classification and indexing.

Accessible educational tools

Recently, there has been an increase in the internet-based delivery of course materials, even when courses themselves are delivered in classrooms. Blackboard (http://www.blackboard.com), for example, provides software and services to schools, colleges, universities, and other educational institutions. While it is involved in various accessibility related projects, such as *Web Accessibility In Mind* (*WebAIM*) and *Standards For Accessible Learning Technologies* (*SALT*), these attempts and projects do not directly address the issue of navigational overload posed on students. Instead, most current technologies aim to make a given single page accessible. Technologies commonly relied upon by the users with visual impairments include screen readers (JAWS, WindowEyes), screen magnifiers (Magnum, ProVi-sion32, ZoomText), voice recognition software, hypermedia-to-hypertext transformers (DragonNS, IBMViaVoice), and refreshable Braille displays (ALVA, PBraille).

Adaptive hypertext and hypermedia

Adaptive hypermedia relies on two different but complementary methods, namely adaptive presentation and adaptive navigation (Brusilovsky, 2001). Adaptive presentation is manipulation of content fragments in a hypertext document. Order of fragments can be changed, or fragments can be made invisible or less visible within a page. Adaptive navigation support, on the other hand, is the manipulation of links. Direct guidance, link sorting, link hiding, link annotation, link generation, and map adaptation are the techniques used. Detailed discussion of all these approaches, both for adaptive presentation and adaptive navigation support, can be found in Brusilovsky (1996, 2001) and Cavanaugh (2002).

Page-accessibility research includes (Huang and Sundaresan, 2000; Mukherjee et al., 2004; Pontelli et al., 2002; Ramakrishnan et al., 2004; Takagi et al., 2002). Takagi et al. (2002) focuses on segmentation and annotation of a given page based on accessible layouts manually predefined using an annotation editor. Huang and Sundaresan (2000) also transforms a given page to render it more accessible. In particular, the transformations may include splitting a single page into multiple units guided through an index. Pontelli et al. (2002) provides a contextual graph for navigation *within* the segments of a given page. Bookmarking and *dialog-based* navigation through page segments are supported in Mukherjee et al. (2004) and Ramakrishnan et al. (2004). As opposed to these techniques, our goal is to provide navigational assistance through a dynamically (i.e., by multiple authors) generated and diversely (i.e., unpredictably) linked collection of information units, such as messages, course pages, and notes.

Researchers in the AI community have developed web navigation tour guides, such as Web Watcher. Web Watcher utilises user access patterns in a particular website to recommend users proper navigation paths for a given topic. Adaptive hyperbooks, such as KBS (Henze and Nejdl, 2000), and guidance systems, like TANGOW (Carro et al., 1999) and ML Tutor (Smith and Blandford, 2003), take into account tasks and user needs, profiles, and access patterns while adapting for learners. Hatzilygeroudis and Prentzas (2004) describes how intelligent tutoring systems can benefit from hybrid knowledge representation formalisms (neurules, which integrate symbolic rules with neural networks based approaches) in classifying users, giving pedagogical decisions, and adapting the teaching material. More specifically, rule conditions are assigned significance factors, while rules are assigned bias factors. Both of these are parameters in the computation of the activation value associated to the rule. In this paper, we also benefit from a rule-based system for achieving modular, incremental classification and annotation of educational material.

Classifying and indexing messages

The problem of classifying and indexing messages in a message board is strongly related to the one of classifying and indexing web documents or pages. One approach to organising web query results based on available web structure is *topic distillation* proposed in Bharat and Henzinger (1998), Chakrabarti et al. (1998) and

Kleinberg (1999). These techniques organise topic spaces as a smaller set of hub and authoritative pages. However, they are usually general purpose and ignore the special hierarchical and dynamic structure of the web content, such as discussion boards. Unlike web documents, postings in discussion boards are in general much more heterogeneous and their informative content is not always structured and easily available. Thus, it is hard (or even impossible), in general, to extract keywords and detect a structure over the messages, and to make them accessible through search engines (Yang and Pedersen, 1997). Furthermore, what users are usually looking for is a conceptual relationship (Candan and Li, 2002) among postings, such as ‘an answer to’, not just keywords. Unfortunately, postings do not embed in the text their explicit relationships with the others.

Xi et al. (2004) creates a specialised ranking function for Usenet. This approach is based on metadata, such as prior knowledge about the message author or the depth of the message. Murakami et al. (2001) suggests a method to extract information from web discussion boards by summarising threads. To extract a thread summary, Murakami et al. (2001) uses quote and comment relationships, which indicate there are topic bindings between messages. Based on the relationships among fragments of text contained in the messages, authors compute distance values among quoted texts in messages. The goal is to relate the quoted parts of different postings to discover their mutual correlation. We expand on this idea and go beyond the simple comparison of quoted text (which, in some cases, does not even exist or does not contain enough informative content, since users are not necessarily required to use quotations) to capture a number of additional elements to recognise the relationships among the postings in a thread. Like Murakami et al. (2001), we use a graph as the data structure to represent the available knowledge.

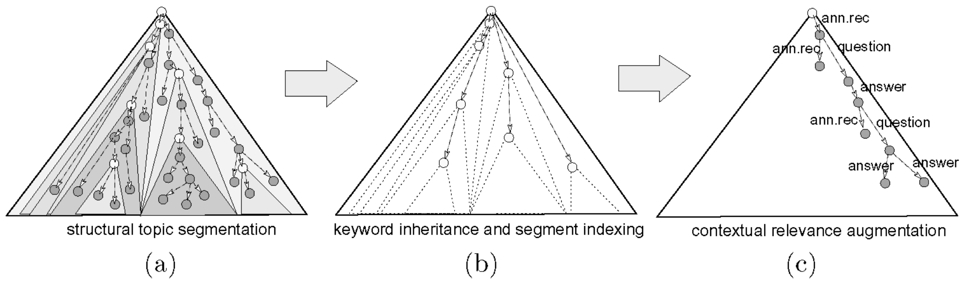
Another approach to classifying message board content has been presented in Aasheim and Koehler (2004). The paper aims at identifying postings about the increase or decrease of a company’s trading volume; therefore, there are only two interesting classes. This approach suggests the use of a formal representation of documents as linear combination of term vectors extracted from the entire corpus of postings. Each term is assigned a frequency and a weight. These are combined to compute a posting score to be used as a discriminator in classification. The intuition of using special terms for analysis is interesting, but the approach has three main limitations. First, each posting is evaluated in isolation from the others; second, the method relies on a set of 15 different parameters to evaluate every single posting, which makes it very complex in practice; and third, the method can not be generalised to other domains or purposes.

In Kim et al. (2005), we focussed on the problem of “searching for special nodes – which are the entry points to new, general, or specific topics – within a single hierarchy of dynamically evolving web content”, with the goal of using this information for segmenting message threads in discussions boards. The core of the approach is to first identify the boundaries of topics, by analysing the keyword vectors of the postings, and then segmenting the topics to discover specialisation/generalisation relationships between directed linked postings. In this paper, we build over this work by introducing an automated off-line system to highlight the topic boundaries and their internal organisation, based on additional structural aspects. In addition to the aspects considered in Kim et al. (2005), we take into account the fact that, even if certain postings might be too short to have a large informative content (and to be characterised in terms of significant keyword vectors (Mladenic and Grobelnik, 1998)), they might still carry information, if considered with respect to some other message. Also, a classification only based on specialisation/generalisation might not necessarily capture the conceptual organisation of the board.

3 Annotation support for navigational guidance

In our previous work, we explored web indexing and mining of web information units (Li et al., 2001a, 2001b), mining document associations (Candan and Li, 2000, 2001, 2002), structural mining of hierarchical content (Candan et al., 2004), and summarisation of websites (Candan and Li, 2002), for sighted people. In Kim et al. (2005), we built on our earlier work by developing segmentation techniques for discovering the topic evolution structures of dynamic and hierarchical web-content, such as discussion boards. In this paper, we extend these results and discuss the application of these techniques for adaptive navigation support to students without sight (Figure 2).

Figure 2 Steps of the segmentation and annotation process. First the discussion hierarchy is segmented; then keywords are propagated among messages as well as segments to properly index each segment; finally, messages in segment are annotated to further help with navigation within and across segments



In this section, we introduce the annotation framework utilised in iCare-Assistant system. Most messages are too short to be meaningful by themselves, and therefore, they obtain their contexts from their parents and ancestors. Thus, the annotation framework integrates topic detection and segmentation mechanisms reported in Kim et al. (2005) as a subcomponent (Figure 2(a)). Once the individual segments are identified, aggregate keyword vectors are computed for each segment. The segmentation step is followed by propagation of keywords in the specialisation/generalisation hierarchies to identify the most suitable keywords to index each segment (Figure 2(b)).

The final, annotation phase (Figure 2(c)) of the process is composed of two complementary steps:

- the first step, devoted to the statistical analysis of the text in postings, leads to the definition of the appropriate classification criteria
- a rule-based system assigns (multiple) classification labels to the messages as annotations.

When interacting with the system, the user then sees the annotated messages. Figure 3 represents this phase in greater detail:

- The *Text Analysis module* performs statistical analysis of the text of the postings in the message board to discover the patterns contained in the different postings. The goal of the analysis is to extract a set of rules that can describe the general behaviour of the authors of the board (Figure 4).

  This module organises the information on which the indexing will be based. First, a *Crawler unit* associates to every message in the board a *message descriptor.* To do so, the Crawler

  - connects to the web server, and fetches all the available messages
  - it eliminates from them the formatting tags possibly appearing and the stop words, and it applies a stemming algorithm and
  - finally, it creates the keyword vectors representing the messages.

  A detailed description of the Crawler activity will be given in Section 4.1.

  In this paper, we consider message boards of a university course discussion forum. In this context, based on the analysis of a few hundred messages, we identified various rules (which we will discuss in Section 5). Each rule is based on a different aspect of the analysed postings (e.g., words used, presence of HTML tagged terms, presence of URL links). The rules codifying the patterns extracted by the domain experts are further analysed with the goal of measuring their extent and validity. The result of this step is an *assignment function*, which associates to every rule a numeric value representing the degree of discrimination it carries.

- Once the vectors, surrogates of the messages, are available, the *Classification module* applies the rules, extracted in the previous step to classify the relationship among different postings, on the basis of their informative content. For this purpose, the postings are formalised as facts populating the working memory of the rule-based system. Nodes are labelled, based on their relationships with other nodes in their thread. Many different reasons may support the relationships between two messages. To take them into account, we associate to every node a list of pair-labels ⟨tag, score⟩; tag qualifies the type of relationship and score is a numeric value which represents the confidence associated to this tag. During the analysis, the labelling can dynamically change: the scores associated to different tags can be updated as the result of the application of the classification rules.

  We represent the discussion board as a tree, in which the nodes represent the postings and the edges represent the relationship between them (Figure 1). The existence of an edge between any pairs of nodes represents the fact that the two messages are consecutive in a message thread.

  Given the University course domain we are considering, this module associates with postings the following tags: *Question*, for postings containing one or more requests on a specific topic; *Direct Answer*, for postings that give a direct answer to any question that had been posted earlier; *Indirect Answer*, for suggestions on where to find the answer to a question; *Announcement/Recommendation*, for postings introducing an announcement, like a news on an event or providing recommendations. Finally, the label *Other* captures postings which have not been classified in any of the previous classes.

Note that the labels provided by the system can always be modified or re-enforced by a system administrator or the user. In our implementation, we do not provide an interface for end-users to choose their own labels.

- The last module of the architecture is the *Message board indexing module*. As a side effect of the classification process, a number of entry points are identified in the message board. These entry points will be the targets pointed by the corresponding topic in the index structure. Therefore, boundaries of the topics will be easily detected during navigation in the graph. Therefore, the navigation is more informed. In particular, it is reduced to a two-level access. First, the external index is used to access a specific topic through its entry point (for example a topic leading with an integrative course of Security, having as entry point a posting titled "An extension to the Security course program"). Then, following the tagged edges (manually or through a query to the system), the user will access the information of interest.

Figure 3 Steps of the annotation process

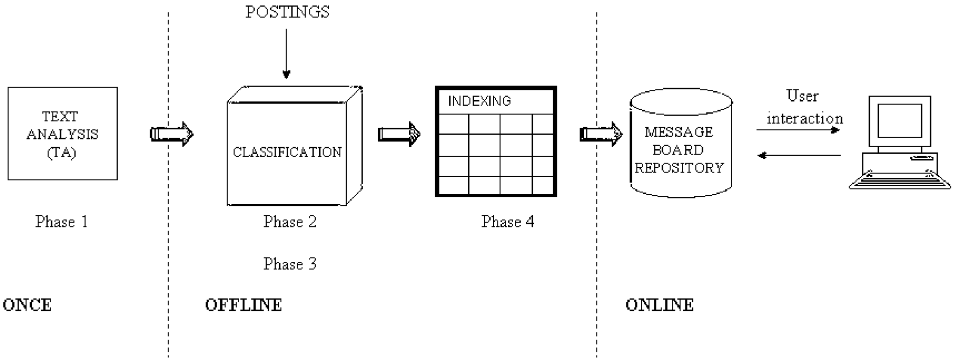
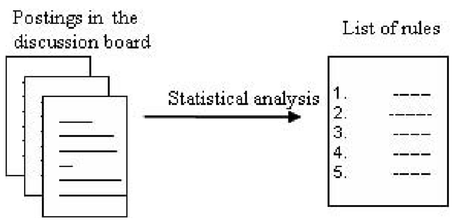


Figure 4 Text analysis process



4 Text analysis module: from messages to message surrogates and segments

In this section, we describe the module which extracts knowledge, in terms of both content and metadata information, from the messages available in the message board. After introducing the role of the *Crawler*, which associates message surrogates to the original postings, we introduce two techniques, the *topic segmentation* and the *keyword propagation* techniques, which we use to enrich the information associated to the message representatives, thus making it more complete, and then more suitable for our classification purposes.

4.1 Crawler module

The *Crawler* unit interfaces directly with the BlackBoard server. It accesses the available information, from which it extracts appropriate descriptors which will be enriched and extended by the next phase of content analysis, to become the input for the classifier module. For every message in the message board, the following steps are executed:

- *Tag elimination*. Formatting tags are removed from the text of the message, so that only the message text is maintained in the message surrogate.
- *Stopwords elimination*. Most frequently used terms (articles, prepositions, etc.) are eliminated, so that only terms which are relevant in the message are maintained. A complete list of stopwords for English language is available at the website http://adsabs.harvard.edu/absdoc/stopwords.html.
- *Stemming*. Morphological and inflectional suffixes are removed from terms. In our implementation we used the stemming algorithm developed by Porter (1980).
- *Keyword vector creation*. For every message, a keyword vector is defined, containing all the terms filtered out by the previous steps.
- *Keyword vector extension*. For every message, an XML document is created. This document contains, in addition to the information available in the keyword vector, several meta-level data. These data will be relevant in the classification of the message, and include: a message identifier, a title, the subject of the message, the name of the author, the date, the length, in terms of number of words in the message, the number of question marks in the message, the quoted text possibly contained within the message, the list of semantically interesting terms, like the ones that characterise an immediate answer (e.g., yes, no, certainly, etc.), or the ones that characterise requests (e.g., what, where, when, etc.), and the description of hyperlinks possibly appearing in the message.

The XML document returned by the keyword vector extension step is the input for the next steps of the analysis and classification of the board.

4.2 Topic segmentation and keyword propagation

As we mentioned earlier, for improving the quality of the annotations, we rely on the message board segmentation techniques developed in Kim et al. (2005). Here we give an overview of these techniques.

Dynamic adaptation of the information space for navigation support requires an indexing system which can leverage the logical relationships between various contents, such as messages that refer to the same assignment within the same context. Messages, on the other hand, obtain their contexts from their parents and ancestors. As a discussion hierarchy grows through posting of new messages, its content and context will also evolve and possibly diverge from the original posting.

In freely (and arbitrarily) evolving message hierarchies in discussion boards, the challenge is not to identify how a document is authored, but to discover how the discussion topics have evolved and how they can be segmented to identify context (topic) boundaries to facilitate indexing, retrieval, ranking, and presentation of appropriate information units (or segments) to the user without sight. Thus, the *segmentation problem* within this context can be defined as searching for special nodes – which are the entry points to new, general, or specific topics – within a single hierarchy of dynamically evolving web content. Once the segmentation is achieved, as in Figure 2(a), each segment can be independently indexed and users can be directed to the entry point of the most relevant segment to their current context.

To *segment a single message chain* we utilise a two-step approach to segmentation: we process the messages in a chain in a top-down manner, and to each message we apply the following steps.

Step 1: *Identifying new topic boundaries*

First, we perform a low-granularity segmentation to identify whether the message is of an unrelated (*new*) topic (relative to the postings immediately before it in the same thread) or not. In a given thread of a discussion board, there is a natural tendency of maintaining the same topic across neighbouring messages, because most postings are replies to previous ones. Thus, a new node does not need to (and cannot) be compared to all its ancestors, but has to be compared to its immediate parent (or an immediate sequence of ancestors) as it is (they are) causally closest to the current node. However, when segmenting discussion threads, there are certain complications. First, consecutive messages of the same topic may be of different length, style, and content. Secondly, in many messaging systems, original postings are automatically included in replies as quotations; hence, unless quotations are used by the author in a way to strengthen the link between the original message and the reply, they may not highlight a common context.

Thus, keywords in quotations should be treated differently based on the *relevance* of the quotations as determined by their placement in the message; in general, quotations selectively used within the body of a message are more relevant than the quotations left (potentially forgotten) as a bulk at the end of a message. Once a keyword weight vector is computed for a message, a threshold on the cosine similarity between this vector and the keyword vector of the parent message (or the keyword vector representing the segment being computed so far) can be used to classify the input message as having a *new* topic or being of the *same topic* as that of the parent.

Step 2: *Segmentation*

If a message is identified to be similar to the previous messages, a *higher-granularity* segmentation process, which tries to determine whether the message is more *specific* or more *general* than the previous messages, is carried out.

Indeed, even though a message may not diverge significantly from the initial theme, it may (a) focus on a specific aspect of the common theme or (b) take the discussion to a more general platform. Finding such boundaries is important because understanding when a discussion topic diverges helps both with indexing (by choosing the right keyword weights for the given segment) as well as guiding the user (without sight) to the most appropriate entry point within a discussion. Therefore, in this step, among the parent/child messages that are identified to be of the same topic, we detect specialisation and generalisation boundaries.

One way to think of generalisation and specialisation is in terms of constraints imposed by the keywords in messages. A message m_1 being more general than m_2 can be interpreted as m_1 being less constrained than m_2 by the keywords they contain. Let us consider two messages, m_1 and m_2, where m_1 contains keywords, k_a and k_b, and m_2 only contains k_a.

- If m_1 is said be more general than m_2, then the additional keyword, k_b of message m_1 must render m_1 less constrained than m_2. Therefore, the content of m_1 can be interpreted as $(k_a \vee k_b)$.
- If, on the other hand, m_1 is said be more specific than m_2, then the additional keyword, k_b must render m_1 more constrained than m_2. Therefore, the content of m_1 can be interpreted as $(k_a \wedge k_b)$.

Note that, in the two-keyword space $\langle k_a, k_b\rangle$, m_1 can be represented by a vector $\langle a_{m_1}, b_{m_1}\rangle$ and m_2 can be represented by $\langle a_{m_2}, 0\rangle$. The extreme point $\mathcal{O} = \langle 0, 0\rangle$, corresponds to the case where a message does contain neither k_a nor k_b; in other words, $\mathcal{O}$ corresponds to a message which can be interpreted as $(\neg k_a \wedge \neg k_b) \equiv \neg(k_a \vee k_b)$. Therefore, if m_1 is said to be more general than m_2, the distance between m_1 and $\mathcal{O}$ should be greater than the distance between m_2 and $\mathcal{O}$. This gives a way to measure and threshold the degrees of generalisation and specialisation of two messages.

Thus, the two-step segmentation process described above is repeated in a top-down fashion, following each chain of the hierarchy independently (Kim et al., 2005). Indeed, two separate replies to a single message are independently created from each other, they can not be marked to be of the same topic unless they are independently identified to be of the same topic as that of their common parent.

Step 3: *Propagation and indexing*

Finally, each connected component of the tree, not split with segment boundaries, is marked as an atomic segment and indexed separately, while the *specialisation* and *generalisation* information is used to identify how keywords are inherited between ancestors and descendants. The common ancestor of all nodes in a given segment is identified as the *entry point* of the segment and used in guiding users.

Once the individual segments are identified, aggregate keyword vectors are computed for each segment (Figure 2(b)). This step is followed by propagation of keywords in the specialisation/generalisation hierarchies to identify the most suitable keywords to index each segment.

5 The rule-based classification module

The goal of the context-sensitive message classification and annotation phase is to extract knowledge about the structural inter-dependencies among messages in such a way that the extracted information can be used for informed navigation or can be directly queried (and thus queries about, for example, the existence of the answer to a specific question can be issued and evaluated).

We aim at partitioning the messages in the board into five classes (Table 1). To do so, we mimic the reasoning patterns adopted by human beings, when they try to classify the messages. A message m can be interesting because of a question it asks (*Question* message, Q), because it strongly confirms or denies some statement in the message it replies to (*Direct Answer* message, DA), because it can be seen as a pointer to a source where the answer to a previous question can be found (*Indirect Answer* message, IA), because of an event it talks about or a suggestion it provides (*Announcement/Recommendation* message). If no one of the above characterisations applies, it is simply classified as an *Other* message (O).

Table 1 Classification types and examples

Classification	*Description*	*Example*
Question (Q)	Postings containing one or more questions on a topic, a request of information, or a request for a solution of a problem	"What is a join algorithm?"
Direct Answer (DA)	Postings containing an answer to a previously posted message in the thread	"It is an algorithm that combines two tables based on a condition"
Indirect Answer (IA)	Postings containing a reference to find an answer for a previously posted message, in terms of pointers to information sources	"Join algorithms are discussed in Section 5 of the book"
Announcement/ Recommendation (AR)	Postings referring to an event (date time, location) or reference to an information source usually not related to a previous question	"Next Friday there'll be an exam at 3PM in BY580" "Course notes for the exam are at http://www.asu.edu/textmining"
Other (O)	Postings that do not fit in any of the previous classes	

It is important to note that even for human beings, the relevant class for a given inter-message relationships is not always uniquely determined. Consider, for example, a message containing the following sentence: "Did you look at the paper http://www.bibliosite/file.pdf?", which might come in reply to a help request. We could have good reasons (the syntactical form, with the question mark at the end of the sentence) to think of the message as a *Question* message. On the other hand, the question can be seen as an implicit way to give an advice: "in case of negative reply, please have a look at the recommended paper", which would support the classification of the message as an Announcement/Recommendation message. Such potential multiple classifications of relationships is one of the motivations for the choice of a rule-based system as the core of our architecture. Rules allow to assign the same pair to multiple classes, with different confidence for distinct classes, and to dynamically adapt the confidence values to take into account the information available at any future point in time. The representative classes for a pair will be the ones with the highest confidence.

The core of the approach consists of an inferential engine working on a rule-based system. In rule-based systems, the reasoning mechanisms typically adopted by human domain experts are coded in a set of if-then-statements, the rules of the system. These rules are used to infer new knowledge, on the basis of the currently available information, expressed by means of a set of initial assertions. Thus, the inferential engine captures, through the rules extracted by the domain expert, the most important features for each posting and makes it easy to classify their role.

5.1 Introduction to rule-based systems

A rule-based system has the following fundamental components

- A set of assertions, which collectively form the working memory, and represent the knowledge/information on which the reasoning is based. The initial working memory is a set of facts which describe the initial data on which the system will base its reasoning activity. In our case, the facts in the initial working memory represent a formal description of the postings contained in the message board, i.e., a logical representation of the message surrogates extracted by the text analysis module.
- A set of *rules* that specify how to act on the assertion set. Rules can be seen as if-then statements, which encode the knowledge of a domain expert, with the goal of reproducing the reasoning schema that a human expert would apply, in the presence of the data currently available, and coded in the working memory. In our case, the rules are used to classify the relationships among different postings (e.g., message m_1 is a reply to message m_2).
- A *termination condition*, to determine that a solution has been found, or to conclude that no solution exists. In our case, the termination condition represents the fact that all currently available postings have been analysed.

The activity of a rule-based system can be described as a loop, which ends when the termination condition is reached. Every iteration of the loop acts through the following steps. First, all the (IF) conditions of the rule are checked, to isolate the set of conditions which are satisfied in the current working memory. If the identified set is empty, the system stops (even if the specified termination condition is not reached yet). Otherwise, from the set of applicable rules (the conflict set), one candidate will be chosen to be fired. The choice of the rule to be triggered, among all the candidates in the conflict set, depends on the conflict resolution strategy associated to the system. In our case, we use the Best rule as the conflict resolution strategy: each rule is given a 'salience', which specifies how much it should be considered over the alternatives, and the rule with the highest salience is chosen. When the selected rule is fired, all the actions specified in its THEN clause are carried out. These actions can have multiple effects, on different targets. In some cases, they simply modify the working memory. In other cases, they also update the rule-base, or execute any actions coded by the system programmer in the THEN clause.

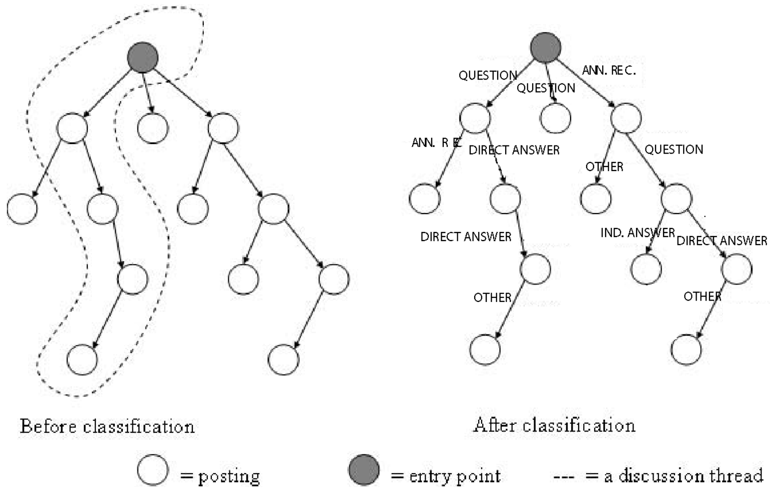
5.2 Structure of the discussion board

In this section we formalise the classification problem and the data structure on which our classification algorithm is based. The *input data structure* for classification is the *discussion tree*, defined as follows.

Definition 5.1 (*Discussion Tree*)**:** The discussion tree is a (non labelled) tree $T = (M, E)$ where $M = \{m_1, m_2, \dots, m_n\}$, the set of vertices, represents single postings populating the message board. $E = \{(m_i, m_j) \mid m_i, m_j \in M\}$ is the set of edges, which connect different inter-related postings. The tree structure of the graph comes from the fact that the only inter-node relationships known at the beginning are the ones induced by the discussion board: if a posting m_j has been written in reply to a previous message m_i, then m_i is the parent of m_j.

Figure 5 is a sample of a discussion tree, before and after the classification. The discussion tree represents the initial structure of the discussion board. Initially, the tree is not labelled. Labels will be added by the classification engine and will qualify the role that the different postings play with respect to the other related postings. While we can assume that the author of a posted message explicitly lists (and maybe quotes) the previously posted messages that he is referring to in his contribution, we cannot expect any explicit contribution to the classification. In principle, the author is not even aware of the classification process being conducted and is not required to declare the meaning of the relationship between the posting he is writing and the one he is replying to.

Figure 5 The discussion tree T, on the left, is before the classification of edges. The tree T' on the right is after the classification



5.3 Classification algorithm

The *classification module* we use is a rule-based system, in which IF-THEN rules are continuously fired (provided their IF conditions is satisfied by the current working memory), until termination.

The *input data structure* for classification is the *discussion tree* of messages. The *output* of the classifier is a labelled tree, the *enriched discussion tree,* in which each node is classified as a *Question*, a *Direct Answer*, an *Indirect Answer*, an *Announcement/ Recommendation,* or an *Other* message. As we already noted, in principle, the same message can belong to more than one of the listed classes, with a different score, representing the degree of membership to the class. During the classification process all different membership degrees are considered and updated. In the classifier module, each classifying rule contributes to the definition/update of the score of the messages. In particular, each rule considers a specific aspect (or a specific set of aspects) of the posting structure.

The choice of a rule-based system, as the classification engine, has the important advantage of allowing a high degree of modularity and ease of reflecting the updates to the message board content. Addition of a new message can change the existing classification as the whole context changes. The *classification module* acts in two steps:

Step 1: (*Content Based Classification*). In this phase, the content of the given messages is taken into account, to provide a preliminary classification. With this goal, a set of content based rules is isolated from the overall set of rules of the classification engine. Different content based rules can give different contribution in the labelling process. A compact representation of the contribution that each rule provide during the labelling process is described by a labelling contribution matrix, defined as follows.

Definition: *Labelling contribution matrix*. Given a set of rules $R = \{r_1, r_2, ..., r_k\}$ and a set of labels $L = \{l_1, l_2, \dots, l_m\}$, a labelling contribution matrix for R and L is a $k \times m$ matrix, ρ, such that $\rho[r, l]$ represents the (absolute or relative) amount that the rule r will contribute to the score of label l when r fires. $\rho[r, l] = 0$ whenever the rule r is not contributing to the score of label l.

For any posting, the existence of a positive score for a given class will be guaranteed if at least one rule in the content based classification module applies to the considered posting.

Intuitively, $\rho[r, l]$ in the labelling contribution matrix can be seen as a parameter that, when the rule r is fired, determines the new score of the label l based on the previously existing score. Therefore, given a segment and a set of rules $R = \{r_1, r_2, \dots, r_k\}$, a set of labels $L = \{l_1, l_2, \dots, l_m\}$, and the corresponding labelling contribution matrix ρ, the scoring function is defined as

$$\text{Score } (l)\,(m_i) = \sum_{r \in R} \rho[r, l] \times C_r(m_i)$$

where m_i is a postings in the message board. Here, $C_r(m_i) \in \{0, 1\}$, is used to distinguish the rules that have been fired ($C_r(m_i) = 1$) from the ones which have not been fired ($C_r(m_i) = 0$).

Table 2 shows a sample rule used in our system. In this specific case, the rule affects the scores of two classes (or annotations), *Indirect Answer* and *Announcement/ Recommendation.* To compute how much each rule should contribute to an annotation (i.e., ρ values used in the rules) the classifier can go through a phase in which training messages are classified and the ρ values are adjusted. In the version of the system evaluated in Section 7, the rules as well as weights on the rules have been adjusted by a human expert.

At the end of Step 1, the result is a partially labelled tree in which only those postings which reach beyond a score threshold for a class are labelled. This partially labelled tree is the input for the first iteration of Step 2 and remaining postings will be labelled during this second step.

Step 2: *Context Based Classification*. Based on the context given by the labelled postings and the segments identified in the earlier phase, messages that are not labelled based on content are classified. At each iteration, for each posting, the class with the highest score is identified and considered as 'the current class' for the posting. In the following iterations, the current classification of each posting will possibly induce an update on the score of other postings' classifications, taking into account the context information. For each posting m in the partially labelled tree

- If the posting m is currently classified as a *Question,* its descendants are analysed, looking for a (direct or indirect) *Answer* to the question. If any *Answer* is found, no further action is taken, and the current classification for the considered message is confirmed.

  If no posting classified as *Answer* exists among the descendants of m, the system tries to find if any descendant, m_o, currently classified as *Other* exists and can be recognised as an *Answer.* If m_o is found, the degree of similarity between m and m_o is computed to estimate how much the two postings are related, and to which degree m_o can be seen as an answer to m. The similarity between postings is calculated based on the *term frequency/inverse document frequency* approach, on the sets of keywords previously extracted from the postings. Both the *Indirect Answer* and the *Direct Answer* scores for m_o are augmented – the actual increase depends on the degree of similarity between m and m_o. It is important to note that a message can be classified as a *Question* based purely on its content and the algorithm does not decrease the probability of a posting to be labelled as a *Question*, if there is no *Answer* descendant found. Yet, if in the context-based classification phase, additional support to the labels are observed, then the score of the label can be increased.

- If the posting m is classified as an *Answer*, whether indirect (IA) or direct (DA), the system looks for an ancestor posting that is classified as a *Question* and might have m as its answer. If found, the current classification for this message is confirmed.

  If no appropriate ancestor *Question* is found, *Other* ancestor messages are considered, to check if any m_o among them could be a candidate as the ancestor *Question*. If m_o is found, its *Question* score is increased, depending on the degree of similarity with m.

  If no m_o is found, the classification of m is updated by decreasing its *Answer* score, since the classification of the posting as an *Answer* is not supported by the presence of a corresponding *Question* in the thread.

- If the posting m is classified as *Other*, the analysis is conducted in both directions, looking for a candidate *Question* among the ancestors and for a candidate *Answer* among the descendants. If no question is found, the systems tries to see if the message m can be related to some *Other* ancestor as a *Recommendation,* based on their similarity.

Step 2 is iterated until there is no change in classification scores; i.e., no further rules can be fired. This is the termination condition. Each iteration of Step 2 can modify the labels of the nodes in the tree as a result of score changes: the update of a score can change the candidate class, that is the class associated to the highest score. There is no commitment about the classification until the entire process is ended. Step 2 is iterated until a fix point is reached, in which scores are not updated anymore. Such a fix point is guaranteed, because the above rules used in context-aware classification do not lead into any circular dependence between classifications.

Postings of new messages in the discussion board result in an incremental application of the rules to relate the new message to the existing ones. The inferential engine underlying the classification system has been developed using Jess, a rule engine and scripting environment written in Sun's Java TM language, inspired by the CLIPS expert system shell. A preliminary version of the classification approach has been discussed in

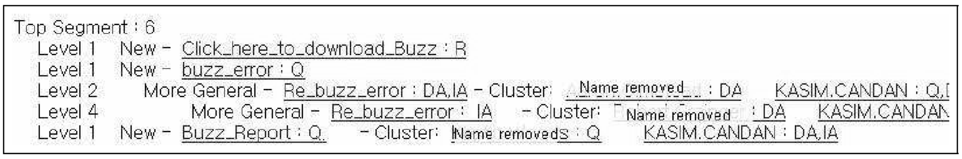
Antonelli and Sapino (2005). The approach, included in this paper, presents a number of major improvement over this basic approach. In particular, in Antonelli and Sapino (2005), the classes used were different from the current ones: all answers were clubbed together, without distinguishing between direct and indirect answers. Moreover, in the preliminary version of this algorithm, classification was performed in one single step. We, however, found that this results in lower classification performance (Antonelli et al., 2005). The two step, content- followed by context-based, classification scheme provides significantly better classification precision. In this paper, we also discuss the integration of the message classification scheme with the segmentation approach to provide context-aware adaptation and annotation of available information.

At the end of the classification process, each message has five different scores, one for each class. Based on these scores, *one or more of the highest scoring classes* are selected for annotating the message (Figure 6).

Table 2 A classification rule example: the firing of the rule states the presence of a hypertext link in the message; *Indirect Answer* and *Announcement/Recommendation* are the two annotations affected by the rule activation

Note: *For this rule,* ρ [*there_is_link, IA*] *= 3 and* ρ [*there_is_link, AR*] *= 3*	
(defrule there_is_link	
	(declare (salience 80))
	?f1 ← (score (post_ID ?id)(type indirect_answer)(value ?v1))
	?f2 ← (score (post_ID ?id)(type ann_rec)(value ?v2))
	(url_link (post_ID ?id))
	(not (done (op there_is_link)(arg1 ?id)))
	(increment (type indirect_answer)(value ?i1))
	(increment (type ann_rec)(value ?i2))
	→
	(modify ?f2 (value (+ ?v2 (* 3 ?i2))))
	(modify ?f1 (value (+ ?v1 (* 3 ?i1))))
	(assert (done (op there_is_link)(arg1 ?id))))

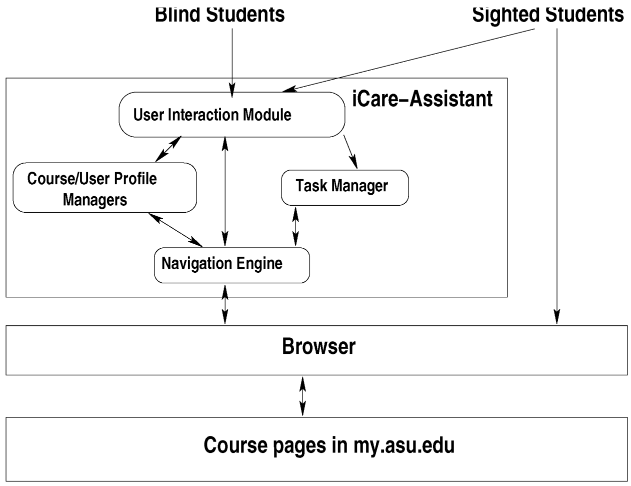
Figure 6 A sample discussion board segment where related messages are clustered based on content similarity. However, within the high-level segment, new topics as well as topic evaluation are highlighted for finer clustering. Also, each message is annotated with one (or more) symbols indicating the type of its content


Top Segment : 6
Level 1 New – Click_here_to_download_Buzz : R
Level 1 New – buzz_error : Q
Level 2 More General – Re_buzz_error : DA,IA – Cluster: Name removed : DA KASIM.CANDAN : Q,[
Level 4 More General – Re_buzz_error : IA – Cluster: Name removed : DA KASIM.CANDAN
Level 1 New – Buzz_Report : Q – Cluster: Name removeds : Q KASIM.CANDAN : DA,IA

6 Usage strategies

The segmentation and context-aware classification algorithms presented in this paper are used for reducing navigational load of the users by adapting the information presentation appropriately. As mentioned earlier, iCare-Assistant provides a transparent interface to the Blackboard system (i.e., the Blackboard system is not modified in any way, but the entire indexing, search, and adaptive presentation schemes are built as a transparent layer on top of the existing functionalities; Figure 7).

Figure 7 Overview of the iCare-Assistant architecture



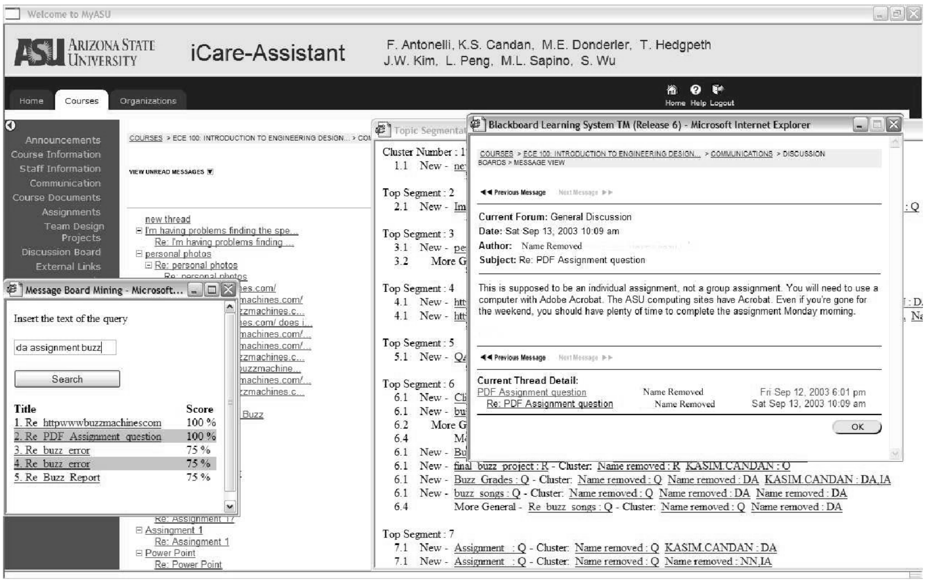
The iCare-Assistant user interaction module is implemented as an HTML Application (HTA) to overcome the security restrictions enforced by the Internet Explorer, and it uses JavaScript to interact with the user and to control the display of the query result. It listens to keystroke events; brings up and presents the most relevant content to the user if a query is given; and allows the user to freely browse course pages on the Blackboard environment.

The user can navigate through course pages as she would do using a web browser, after logging onto the MyASU website. The user interaction module of the iCare-Assistant is designed to be fully accessible through keyboard and JAWS for blind students. It is transparent to the user in that it feels as if the blind user were accessing the Blackboard system through a web browser only, but with the query and adaptation functionalities added to locate the most relevant course content with a few interactions, which the Blackboard system lacks.

The user launches the query dialog by keystrokes. After a query is submitted, the page that contains the most relevant information to the user's query is found, loaded, and displayed on the content frame with the information unit determined to be the best answer at the top. JAWS speaks the returned content to the blind user starting from the top of the content frame, where the query answer (information unit) is located, rather than reading from the beginning of the loaded page (default action in JAWS) which would be done only if the page is the best answer to the query as an information unit itself. The navigational load is minimised by bringing up the most relevant content to blind students from the hyperlinked course information space with only a few interactions by the user. The blind user does not need to have a specific training to learn how to use the iCare-Assistant, except for remembering a few key combinations to launch the query dialog and for the predefined topic-specific retrieval tasks.

Students can either perform a keyword query (through a keyword entry form that pops-up when the user selects a special hot-key combination) or can browse the annotated discussion board. These access mechanisms are presented in Figures 6 and 8. Below we discuss different use patterns for iCare-Assistant.

Figure 8 Sample search for a relevant *direct answer*: the context is provided by the user through the string ‘da’ preceding the search keywords; ‘da’ denotes that the user is interested only in ‘direct answers’ relevant to the query terms. The figure also shows the content of a matching message selected from the list provided by the iCare-Assistant



Informed browsing and search

Figure 6 presents the first usage strategy. In this strategy, the system segments the message hierarchy and clusters the messages in a single segment together. This clustered hierarchy is presented to the user as an alternative navigation tool. In order to further help the user with the navigation, each message in the clusters is also annotated with its context-aware classifications. When there is a clear winner in the system classification, the users are presented with a single annotation; on the other hand, when multiple classifications of the same message are likely based on the scores, the system presents alternative annotations to help the user make informed navigation decisions (Figure 6). When the user provides a search criterion for accessing the discussion board, the system directs the user to the relevant cluster of messages. The user can browse within the relevant cluster using the provided annotations.

Focussed search by annotation

In the second, *focussed search*, use strategy, the user specifically asks for a *question*, a *direct answer*, a *recommendation*, etc. related to a particular set of keywords. In this case, the system limits the results to the specific type of message the user is looking for. For instance, in Figure 8, by specifying ‘DA’, the user focuses the attention to relevant *direct answers.*

7 Experimental evaluation

In order to evaluate the effectiveness of the annotation techniques presented in this paper, we performed user studies, and compared the annotation performance of the proposed algorithms with the feedback provided by assessors.

Evaluators

Our primary goal in the evaluation was to judge whether it is possible to annotate messages in away that will correctly inform users. Since the label of a message (e.g., whether a message is a question or not) does not depend on whether the user is blind, the evaluators did not need to be blind. In fact, since users who are blind have difficulty accessing discussion boards which do not provide support, having evaluators with visual impairments could impose negative bias against messages that are too deep as such users may loose the context information that is available to those users with vision. Since our overall goal is to learn the labels that a person with vision could deduce and provide these labels to users who are blind, a successful labelling scheme has to reflect the judgments of users who have access to visual information. Thus, we used eight assessors[1] with vision to provide label judgments. The number of assessors was chosen to reflect the number of users in similar studies in the computer human interaction literature (Bellotti et al., 2003; Kaptelinin, 2003). However, in order to be sure about the statistical relevance of the results obtained through the user study, we also ran statistical tests to judge the sufficiency of the observations. The results of the statistical tests are also reported below.

Evaluation setup

Since the type of content is important for the evaluation of the classification and annotation schemes presented in this paper, we used a course discussion board available at `my.asu.edu`, with 61 messages, at Arizona State University. The assessors involved in the study were provided with a questionnaire which

- provides the definitions of the different classes
- presents one example per each class
- asks them to classify each message in the board in *only one* of the five classes, based on the message's context within the thread it belongs to.

Assessors taking part in the study had been informed about the meaning we assign to different labels, but they did not know the underlying scoring mechanism. Also, to prevent any bias in their feedback, evaluators did not have access to the automatically labelled version of the message board. They had access only to the original message board and they were asked to provide labels based on their own judgements.

Evaluation results

To evaluate the effectiveness of the classification scheme under a very stringent success definition, we choose *one* class (the one with the largest vote from the users) as the *user-class* and *one* class (the one with the largest score) as the *system-class.* To evaluate the actual correspondence between these user and system classifications, we defined the precision of the system as the ratio of the matches between the user and system classifications. The user study results showed that the system annotations are well aligned with the annotations provided by the students, leading to a 0.89 precision overall. Table 3 shows the classification matrix (user classification vs. system classification) obtained through the user study. Note that most errors are among similar categories. The corresponding success rates for each individual class are also high:

Q: 1.00, IA: 0.80, DA: 0.83, AR: 0.88, and O: 0.78.

Table 3 User classification vs. system classification

User/system	*Q*	*DA*	*IA*	*AR*	*O*
Q	*19*	0	0	0	0
DA	0	*10*	1	0	1
IA	0	0	*4*	1	0
AR	1	0	1	*14*	0
O	1	1	0	0	*7*

In order to evaluate the statistical significance of the results presented above, we ran two different statistical tests on the data. We first ran an ANOVA test (Julian Faraway, 2002), which measures the difference between the means of two or more groups (evaluator label judgements in our case). It is generally used to measure if there is a significant difference between members of a group. The ANOVA test provides an F value which needs to be smaller than an F_{critical} value to ensure that none of the observations is significantly different from the others. For our experiments, the F value was 0.64 against a F_{critical} value of 2.03, highlighting the fact that the evaluator judgments were sufficiently correlated to provide a 95% confidence to the results.

We also used a technique called two-step sampling (Haag and Tonn, 1997) to judge the number of necessary evaluators. In this test, first a small initial number of evaluators are used to collect an initial sample. Then, t-values of this initial sample are used to judge how many more samples (evaluators) would be needed to be 95% certain that the mean precision is within ±0.03. For our experiments, when the two-step sampling technique is utilised, the number of required samples were identified as 7. Thus, our experiments use one more person than needed for being statistically significant.

In addition, we also evaluated whether the sample of messages we used in the experiments were sufficient for statistical significance of the precision results reported for the labelling algorithm. For this purpose, we used Fisher's Z-transformation (Julian Faraway, 2002) which judges whether a given sample of observations (the messages in this case) can provide >95% confidence on the correlations that are observed. For each of the labels, this test showed that, indeed, we can be more than 95% confident of the observed correlations between user and the system judgments.

8 Conclusions

In this paper, we presented the annotation techniques we developed to enable navigational helps for individuals who are blind when using educational discussion boards. The complementary segmentation and annotations algorithms are being deployed in a software system, called iCare-Assistant, which aims at reducing the navigational load for blind students in accessing web-based electronic course materials through an unobtrusive, task-oriented, and individualised delivery interface. The tool is being used by the non-sighted students who are getting their Computer Science and Engineering degree at ASU. Our future work includes extending the navigation technologies to different types of educational material and leveraging the course content and user focus to greater extend to improve the precision.

Acknowledgements

We thank our students without sight, Kenneth Spector and David Paul, for their feedbacks, and Terri Hedgpeth for her expertise in assistive technology.

This is an extended version of Antonelli et al. (2005) originally published in *Proc. CIAH/WBIES 2005 – Combining Intelligent and Adaptive Methods/Techniques in Web Based Education Systems*, 2005. This work is supported by NSF ITR Grant, ITR-0326544.

References

Aasheim, C. and Koehler, G.J. (2004) 'Mining message board content on the world wide web for organizational information', in Barko, C. and Nemati, H. (Eds.): *Organizational Data Mining: Leveraging Enterprise Data Resources for Optimal Performance*, Idea Group, Inc., pp.188–200.

Antonelli, F. and Sapino, M.L. (2005) 'A rule based approach to message board topics classification', *Proc. MIS05*, *LNCS*, Vol. 3665, pp.33–48.

Antonelli, F., Kim, J.W., Candan, K.S. and Sapino, M.L. (2005) 'Navigation support for students who are blind in accessing discussion boards', *Proc. Combining Intelligent and Adaptive Methods/Techniques in Web Based Education Systems*, Salzburg, Austria, pp.1–10.

Bellotti, V., Ducheneaut, N., Howard, M.A. and Smith, I.E. (2003) 'Taking email to task: the design and evaluation of a task management centered email tool', *ACM Conference on Human Factors in Computing Systems (CHI'2003)*, April 5–10, Fort Lauderdale, FL, ACM, NY, pp.345–352.

Bharat, K. and Henzinger, M. (1998) 'Improved algorithms for topic distillation in a hyper-linked environment', *21th Annual International ACM SIGIR Conference*, Melbourne, Australia, pp.104–111.

Brusilovsky, P. (1996) 'Methods and techniques of adaptive hypermedia', *User Modeling and User-Adapted Interaction*, Vol. 6, Nos. 2–3, pp.87–129.

Brusilovsky, P. (2001) 'Adaptive hypermedia', *User Modeling and User-Adapted Interaction*, Vol. 11, No. 1, pp.87–110.

Candan, K.S. and Li, W.S. (2000) 'Using random walks for mining web document associations', *Pacific Asian Conf. on Knowledge Discovery and Data Mining*, Kyoto, Japan, LNCS 1805, pp.294–305.

Candan, K.S. and Li, W.S. (2001) *Discovering Web Document Associations for Web Site Summarization*, DaWaK, LNCS 2114, Munich, Germany, pp.152–161.

Candan, K.S. and Li, W.S. (2002) 'Reasoning for web document associations and its applications in site map construction', *Int. Journal of Data and Knowledge Eng*., Vol. 43, No. 2, pp.121–150.

Candan, K.S., Kim, J.W., Liu, H. and Suvarna, R. (2004) 'Structure-based mining of hierarchical media data, meta-data, and ontologies', *5th Workshop on Multimedia Data Mining in Conjunction with the ACM SIGKDD.*

Carro, R.M., Moriyon, R., Pulido, E. and Rodriguez, P. (1999) 'Teaching tasks in an adaptive learning environment', *Human-Computer Interaction. Communication, Cooperation and Application Design*, Vol. 2, pp.740–744.

Cavanaugh, T. (2002) 'The need for assistive technology in educational technology', *Educational Technology Review*, Vol. 10, No. 1, pp.27–31.

Chakrabarti, S., Dom, B., Gibson, D., Keinberg, J., Raghavan, P. and Rajagopalan, S. (1998) 'Automatic resource compilation by analyzing hyperlink structure and associated text', *World Wide Web Conf.*, Brisbane, Australia, pp.65–74.

Cohen, W.W. (1996) 'Learning rules that classify e-mail', *AAAI Spring Symposium on Machine Learning in Information Access*, pp.18–25.

Donderler, M.E., Candan, K.S., Wu, S., Peng, L. and Kim, J.W. (2004) 'Adaptive electronic course content delivery for students who are blind', *Demo. at ACM Conf. on Computers and Accessibility (ASSETS04)*, Atlanta, GA, USA.

Donderler, M.E., Peng, L. and Candan, K.S. (2003) 'Adaptive content delivery to assist blind students in accessing course materials', *Accessing Higher Ground Conf. Assistive Tech. and Accessible Media in Higher Ed.*, Boulder, Colorado.

Haag, M. and Tonn, W.M. (1997) 'Sampling, density estimation and spatial relations', in Karcher, S.J. (Ed.): *Tested Studies for Laboratory Teaching*, The University of Calgary, June 10–14, Vol. 19, pp.197–216.

Hatzilygeroudis, I. and Prentzas, J. (2004) 'Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities', *Journal of Expert Systems with Applications*, Vol. 26, No. 4, pp.477–492.

Henze, N. and Nejdl, W. (2000) *Extendible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material*, AH, Trento, Italy.

Huang, A. and Sundaresan, N. (2000) 'A semantic transcoding system to adapt web services for users with disabilities', *ASSETS*, Arlington, VA, pp.156–163.

Julian Faraway (2002) *Practical Regression and Anova using R*, http://www.stat.lsa.umich.edu/~faraway/book.

Kaptelinin, V. (2003) 'UMEA: translating interaction histories into project contexts', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Ft. Lauderdale, Florida, USA, April 05–10, 2003)*, CHI'03, ACM Press, New York, NY, pp.353–360.

Kim, J.W., Candan, K.S. and Donderler, M. (2005) 'Topic segmentation of message hierarchies for indexing and navigation support', *WWW 2005*, May, Chiba, Japan, pp.322–331.

Kleinberg, J. (1999) 'Authoritative sources in a hyperlinked environment', *Journal of the ACM*, Vol. 46, pp.604–632.

Li, W.S., Candan, K.S., Vu, Q. and Agrawal, D. (2001a) 'Query relaxation by structure and semantics for retrieval of logical web documents', *IEEE TKDE*, Vol. 14, No. 4, pp.768–791.

Li, W.S., Candan, K.S., Vu, Q. and Agrawal, D. (2001b) 'Retrieving and organizing web pages by information unit', *WWW Conf.*, Hong Kong, pp.230–244.

Liu, B., Chin, C.W. and Ng, H.T. (2003) 'Mining topic-specific concepts and definitions on the web', *WWW Conf.*, Budapest, Hungary, pp.251–260.

Mladenic, D. and Grobelnik, M. (1998) 'Feature selection for classification based on text hierarchy', *Working Notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, Pittsburgh, PA.

Mukherjee, S., Ramakrishnan, I. and Kifer, M. (2004) *Semantic Bookmarking for Non-visual Web Access*, ASSETS, Atlanta, GA, pp.185–192.

Murakami, A., Takeda, K. and Nagao, K. (2001) *Discussion Mining: Knowledge Discovery from Online Discussion Records*, NLPRS Workshop XML and NLP.

Pontelli, E., Gillan, D., Xiong, W., Saad, E., Gupta, G. and Kashmer, A. (2002) *Navigation of HTML Tables, Frames and XML Fragments*, ASSETS, Edinburgh, Scotland, pp.25–32.

Porter, M. (1980) 'An algorithm for suffix stripping', *Automated Library and Information Systems*, Vol. 14, No. 3, pp.130–137.

Ramakrishnan, I.V., Stent, A. and Yang, G. (2004) *Hearsay: Enabling Audio Browsing on Hypertext Content*, WWW, New York, pp.80–89.

Smith, A.S.G. and Blandford, A. (2003) 'ML tutor: an application of machine learning algorithms for an adaptive web-based information system', *IJAIED*, Vol. 13, pp.235–261.

Takagi, H., Asakawa, C., Fukuda, K. and Maeda, J. (2002) *Side-wide Annotation: Reconstructing Existing Pages to be Accessible*, ASSETS, Edinburgh, Scotland, pp.81–88.

Xi, W., Lind, J. and Brill, E. (2004) *Learning Effective Ranking Functions for Newsgroup Search*, SIGIR, Sheffield, UK, pp.394–401.

Yang, Y. and Pedersen, J.O. (1997) 'A comparative study on feature selection in text categorization', *Proc. Fourteenth International Conference on Machine Learning*, San Francisco, CA, pp.412–420.

Note

[1]Assessors were not involved in the annotation project and were not familiar with the sample discussion board; therefore, for the purposes of the experiment, they were naive users seeing and interpreting the discussion board for the first time.

Website

Blackboard http://www.blackboard.com.