This document is published in:

Delgado Kloos, C. et al. (eds.) (2012). *Towards Ubiquitous Learning: 6th European Conference of Technology Enhanced Learning, EC-TEL 2011, Palermo, Italy, September 20-23, 2011. Proceeding*, (Lecture Notes in Computer Science, 6964), Springer, pp. 327-340.
DOI: 10.1007/978-3-642-23985-4_26

Automatic Discovery of Complementary Learning Resources

Vicente Arturo Romero Zaldivar[1,2], Raquel M. Crespo García[2], Daniel Burgos[1,3], Carlos Delgado Kloos[2], and Abelardo Pardo[2]

[1] Atos Origin SAE,
Albarracín 25, 28037 Madrid, Spain
{vicente.romero,daniel.burgos}@atosresearch.eu
[2] Department of Telematic Engineering,
University Carlos III of Madrid, Spain
{rcrespo,cdk,abel}@it.uc3m.es
[3] Universidad Internacional de La Rioja,
Gran Vía Rey Juan Carlos I 41, 26002 Logroño, Spain

Abstract. Students in a learning experience can be seen as a commu-nity working simultaneously (and in some cases collaboratively) in a set of activities. During these working sessions, students carry out nu-merous actions that affect their learning. But those actions happening outside a class or the Learning Management System cannot be easily observed. This paper presents a technique to widen the observability of these actions. The set of documents browsed by the students in a course was recorded during a period of eight weeks. These documents are then processed and the set with highest similarity with the course notes are selected and recommended back to all the students. The main problem is that this user community visits thousands of documents and only a small percent of them are suitable for recommendation. Using a combination of lexican analysis and information retrieval techniques, a fully automatic procedure to analyze these documents, classify them and select the most relevant ones is presented. The approach has been validated with an em-pirical study in an undergraduate engineering course with more than one hundred students. The recommended resources were rated as "relevant to the course" by the seven instructors with teaching duties in the course.

Keywords: Personalisation, recommendation, adaptive mentoring, learning analytics, information retrieval.

1 Introduction

This paper focuses on how students search for useful resources in Internet while participating in a learning experience. Searching for information is a task commonly required in course activities of any kind (distant, face-to-face or blended learning). Even if a set of exhaustive course notes is produced, students still search for additional information in Internet. Thus, a community of students emerges that is simultaneously visiting a potentially large number of resources,

some of which could be relevant to the course. These resources could be automatically detected by analyzing this process. But there are two major hurdles to analyze the activity of such community. First, the resources viewed by the students are not trivial to monitor without the student intervention (for example, with a rating system). And secondly, from the visited resources, only those truly relevant for the course should be detected and eventually reported back to the community.

Current Learning Management Systems (LMSs) typically store usage and event information in databases and log files that can later be accessed and analyzed using data mining techniques [12]. Students, however, tend to use certain tools that lie outside the scope of the LMS[16,19]. Skills commonly included in courses such as "managing various information sources", or "search for relevant information", are developed mainly in the user personal space. This type of activities are difficult to supervise both in remote as well as in face-to-face learning.

When a student starts working in a course activity, it is highly likely that she searches the Web for information that is not present in the course resources or perhaps not adequately explained. This behavior may be originated by the need of solving a very specific problem or in other cases by the need of obtaining a tutorial type of document explaining step by step how to solve a given problem. During this process, students may encounter resources related to the course that could be interesting for their peers.

The main question addressed in this document is whether the resources viewed by the students (apart from conventional course notes) can be automatically processed to obtain a set of resources related to the course.

Ideally, the procedure by which these additional resources are identified should be executed without student intervention. Although students clearly embraced the Web 2.0 philosophy [2], current state-of-the-art technologies such as information retrieval and data-mining show that useful information can be automatically derived by processing user observations.

These techniques can be used to automatically detect similarities between documents without manual classification or labeling. The idea in this work is to create a reference corpus with the course documents, and use it to determine how additional resources visited by the students in Internet are related to them. Those resources that are similar to the course material can be recommended to the students. The steps to obtain these recommendations are: track the student browsing activity, process the set of obtained URLs, characterize the course resources offered by the teaching staff, compare both document sets, and determine which additional resources are most related to the course.

The rest of the paper is organized as follows. Section 2 discusses related work and technologies used in the presented approach. The adopted solution is described in Section 3. An empirical study followed a statistical valuation is presented in Section 4. Finally, conclusions and future work are discussed in Section 5.

2 Related Work

The work described in this paper is related to the following areas: user monitorization, recommender systems, and information retrieval.

Monitorization of student activity is typically done by analyzing the logging facilities included in LMSs or other learning platforms (e.g. [18]). There are numerous applications in which this information is used to anticipate student behavior and take some corrective actions (see [12] for an example). But the trend in learning environments is toward using a variety of tools among which the learning activitites are distributed. The tendency is for users to design their own Personal Learning Environment by combining multiple applications [3]. This increase of the activities outside the LMS is particularly high when methodologies such as problem-based or project-based learning are applied. Complementary mechanisms for gathering information about such activities would thus be useful for improving the understanding of the student's learning process.

In the field of Technology Enhanced Learning, monitoring student activity has been proposed as a tool for promoting self-reflection [10,22], awareness [10,14], student assessment [14], course management [15], course evaluation [25], and, of course, learning adaptation and personalization [26], among other potential applications.

Attention.XML [24,6] was an early approach to capturing and storing attention metadata, that is, to represent attention. Due to its insufficient granularity and the lack of context information, the CAM (Contextualized Attention Metadata) Schema has been defined as an extension of Attention.XML [26], focusing the most important extensions on actions that occur on data objects [23]. The procedure described in this paper is based on monitoring student's web surfing activity in order to gather a collection of potentially interesting resources to complement course materials.

The number of learning resources available now to students has increased massively. As it has been the case in other fields, such increase prompts the need for improved methods to find and retrieve these resources [13]. Recommender systems were originally defined as those in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients" [17]. The term now has a broader connotation, describing any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options [5].

Recommendation techniques are classified into two main categories: model-based and memory-based [1]. As explained in [9], model-based techniques periodically cluster the data in estimated models, using techniques such as Bayesian models, neural networks or latent semantic analysis. Memory-based techniques continuously analyse all user or item data to calculate recommendations. The latter can be further classified into collaborative filtering, content-based and hybrid. Collaborative filtering techniques follow a social approach, recommending

items based on the preferences of similar users. Content-based recommendation systems try to recommend items similar to those a given individual user has liked in the past [11].

In this paper, the content-based approach has been adopted. The course material provided by the teaching staff is used as the input information for characterizing student interests and needs. Resources visited by the students are then evaluated against this set of relevant items in order to identify their adequateness for course purposes.

This adequateness is obtained using Information Retrieval techniques, more concretely, the algebraic vector space model proposed by Salton in [21]. In this model each document is characterized by a vector of term weights. Each vector component represents the weight of the corresponding term in the document and is calculated based on local and global parameters. Such model is known as term frequency-inverse document frequency (TF-IDF), and computes the weight of a given term based on the frequency of the term in the document itself and in the complete collection. By weighting the local term frequency with the global inverse document frequency, the TF-IDF model avoid associating high values to common terms that do not characterize the document when compared to the rest of the collection, despite potentially high local frequencies in the document. Once modeled by vectors, the similarity between two documents (or between a document and a query) can be easily calculated using techniques such as the cosine of the angle between the vectors.

Information retrieval algorithms are complemented in this work with natural language processing, in order to achieve a more accurate set of descriptive terms. In this field they are several tools than can be used for example GATE [7] which is a General Architecture for Text Engineering. GATE, according to its authors, is intended to apply to whole families of problems within the language processing field, and to be like a toolbox in service of construction and experimentation. GATE provides not only the algorithms to analyze text but also tools for the visualization of the results which is of great help in the process of discovering similarities between documents.

In this paper, we use information retrieval techniques combined with natural languages algorithms to process a set of resources visited by the students while working on a course and select the most relevants so that they can be recommended to the entire course. The whole procedure is done automatically and requires no interaction with the students.

3 Detecting Resources Outside the LMS

Learning experiences are including an ever increasing number of resource types: documents, videos, audio clips, Web pages, etc. In principle, these resources are published by the instructors so that students use them to carry out the course activities. But in the information age, it is not uncommon for activities to require *explicitly* the search for additional resources. Even in courses where the resources made available cover exhaustively the considered topics, when faced

with any course task, students resort to Internet to find additional information. This search for information is spreading both to individual and collaborative activities. This paper proposes a technique to automatically select the most relevant resources from those visited by the students while working on a course.

These *unofficial* resources visited by the students may complete or explain with other words any of the course concepts. Sometimes students find in these resources solutions to tasks related to those required in the course material. Additionally, students (specially digital natives) are used to look for tutorials and hands on documents explaining how to solve all types of problems. Independently of the pedagogical strategy used, students tend to solve problems using not only the course documentation but also, and in a significant percentage, external resources available in Internet. Ideally, all these searches and discoveries could be collected and processed to select a subset of "most relevant" resources and recommend them back to all the students in a course. Furthermore, the procedure could be deployed transparently to the students with no need for rating schemes.

The generic scenario considered for this paper is a learning experience where a set of resources previously selected (or produced) by the instructors are made available to the students. When working on the course activities, students search for auxiliary material in Internet. These additional visited resources are then collected and analyzed. It follows a description of the strategies adopted in a real case that fits into this generic scenario.

3.1 Data Collection Strategy

The main challenge when collecting the sites that a student visits while carrying out activities related to a course is precisely to restrict the observation only to that situation. In a typical scenario, students use a personal computer in which *at certain times*, they use the browser to check the course material and maybe search for additional material. Access to course material is typically monitored using the logs stored in the LMS hosting the course. But the focus of this study is to quantify and analyze those sites that are visited outside the scope of the LMS while working on a course.

The adopted strategy consisted on creating a virtual machine (a file containing an entire computer that can be run as another application in a personal computer) offered to the students at the beginning of the course. The virtualization platform used was VirtualBox[1]. The advantage for the students is that this machine contained a fully configured working environment including all the additional tools required in the course. This machine was instrumented so as to store and then rely to a remote server the URLs that were introduced in its internal browser as described in [16]. To comply with personal data privacy legislation, students are informed of this instrumentation as well as the procedure to deactivate it if they wish to do so. Additionally, students were advised by the instructors at the beginning of the semester to restrict the work in this machine only to activities directly related with the course. This strategy offers a

[1] www.virtualbox.org

reasonable trade-off between a monitoring scheme that would record every single action on their personal computers, and a system that can only analyze their interaction with the LMS.

When a student boots the virtual machine *and* its internal browser is used, every time a new URL is opened, a line containing the URL and the time stamp is stored and later relayed to a remote server. As a result, the set of URLs visited by all the students (that used the virtual machine and decided to maintain enabled the recording mechanism) is collected. Separating the URLs pointing to the course material is simple because their location is known in advance. The objective, then is to automatically analyze the remaining URLs and select those pointing to resources that are most similar to the course notes, thus maximizing the probability of detecting useful complementary material.

3.2 Calculating the Similarity

The proposed algorithm receives as input two sets of URLs: those obtained from the students, henceforth called the "student set", and those pointing to the course resources or "reference set". The steps in the algorithm to process these sets are depicted in Figure 1.

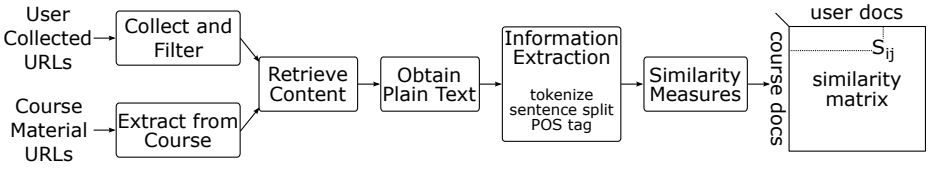

Fig. 1. Data processing steps to compute the similarity matrix

An initial filtering step is applied to the user collected data to remove unwanted URLs. For example, URLs pointing to course material are not considered because they are already part of the reference set. Additionally, a fixed set of domains corresponding to social networks, email services, etc. must be removed as they provide irrelevant information for the detection process[2]. The analogous process in the reference set amounts simply to extract the URLs pointing to the course resources from the proper source (the official course material repository, a list of all the handouts, etc.)

During the second step the content is automatically retrieved from Internet and classified attending to the type of resource (HTML page, image, PDF document, plain text, Microsoft Word, etc.) This is done to select only resources that can be translated to plain text to apply lexical analysis and information

[2] Although occasionally discussions related to the course can happen in these domains, they are password protected and thus private beyond the scope of the analysis.

extraction techniques. During the translation to plain text, the algorithm applies different tools depending on the type of the resource and both sets are then translated into plain text documents.

The information extraction step is carried out using Gate, a toolkit for natural language processing [7]. First, the documents of both the student and reference sets are used to create and populate a "collection" or "corpus". This new set is needed by the Gate Document Manager to perform the text analysis and obtain the proper annotations. Although Gate offers some support for processing non-plain documents, mainly in XML format, due to the presence of multiple formats and sometimes incorrect HTML documents in the student set, the translation to plain text was performed by an ad hoc step.

Once the corpus is created, an automatic sequence of so-called "processing resources" is applied to all the documents. Gate includes a set of algorithms for natural language processing called "ANNIE" (A Nearly New IE System) consisting of tools such a tokenizer, a sentence splitter or a Part-of-speech (POS) tagger, etc. (see [8] for a more detailed description). The application discussed in this paper uses these tools to obtain an annotated view of each document where nouns and proper nouns are identified.

The input data for the last stage shown in Figure 1 is a set of annotated documents in XML in which the nouns and proper nouns have been identified. The output is a matrix M of size $n \times m$ where n and m are the number of documents in the student and reference sets respectively and such that M_{ij} is how similar document D_s from the student set is to the reference document D_r.

The similarity coefficients are obtained using ranking techniques conventionally used in information retrieval [4]. First, each document is translated into an n-dimensional vector where n is the number of terms (nouns and proper nouns) identified in the previous step. Two documents can be compared by computing the cosine of the angle of their corresponding vectors. Although there are numerous well established and robust techniques to compute this coefficients, some specific features of the considered context need to be taken into account.

Typically, ranking systems solve the problem by which given a set of documents and "a query", the subset of most relevant documents for that query needs to be computed. The scenario discussed in this document is slightly different. The input data are two sets of documents, but one of them, the reference documents, limit the scope in which new relevant resources need to be discovered. In other words, given the set of reference documents, select those documents from the student set that are more similar.

Similarity measures and ranking techniques have been widely studied in the area of information retrieval. The selected approach uses term weighting techniques that combine the distribution of a term within a document, but also within a collection. However, in order to force the algorithm to detect documents relevant to the course topic, for computing both the vocabulary and the weight of a term within the collection, only the reference documents are considered. With this adjustments, given the vectors $D_s = (w_{s1}, \ldots, w_{st})$ and $D_r = (w_{r1}, \ldots, w_{rt})$

representing two documents in the student and reference sets respectively, the similarity coefficient is computed as follows [21]:

$$Similarity(D_s, D_r) = \frac{\sum_{i=1}^{t}(w_{si} \times w_{ri})}{\sqrt{\sum_{i=1}^{t}(w_{si})^2 \times \sum_{i=1}^{t}(w_{ri})^2}}$$

The set t contains all the terms identified by processing *only* the nouns in the reference documents. The weights of each term for each of the documents (w_{ri} and w_{si} standing for the weight of term i for a document in the reference or student set, respectively) have been calculated following the $tf*idf$ product [20]. The document vectors are obtained as follows:

$$\begin{aligned} w_{ri} &= tf_{ri} * idf_{ri} \\ w_{si} &= tf_{si} * idf_{ri} \\ tf_{ri} &= freq_r i/|terms_r| \\ idf_{ri} &= \log_2 \frac{N - n_i}{n_i} \\ tf_{si} &= 0.5 + \frac{0.5 freq_{si}}{max freq_s} \end{aligned}$$

where N is the number of documents in the reference set, n_i is the number of documents in the reference set where term i appears, $freq_{ri}$ and $freq_{si}$ are the appearances of term i in a reference or student document respectively, $|terms_r|$ is the number of terms in a reference document, and $max freq_s$ is the maximum number of appearances of a term in a student document.

The similarity values are numbers between zero and one where zero means maximum distance and one means minimum distance. It thus provides a metric for measuring the relevance of each resource related to the course topic(s), automatically filtering out the non-relevant ones and allowing to selecting the closest ones for recommendation.

4 Empirical Study

In order to validate the hypothesis enunciated in Section 1, a monitoring mechanism was deployed in a face-to-face, second year engineering course with 220 students over an 8 week period. The course topic is C programming but students are supposed to use additional tools such a memory profiler, debugger, integraged development environment, and a version control system. The cohort is divided into sections of 40 students that meet twice a week. An active learning methodology is used by which students need to work on activities *before* each class. The virtual machine described in Section 3 contains all the required tools to work in these activities.

The course lasted for 14 weeks in the Fall semester of 2010, but the URL recording mechanism was used only for the initial 8 weeks. The reason for this time window is to show that, if the approach is successful and a set of meaningful auxiliary resources is obtained automatically, they can be incorporated into the on-going course edition. Table 1 summarizes the information obtained during this period.

Table 1. Summary of the data collected in the study

Students participating	125
Total number of URLs collected	17,787
URLs pointing to course material	6,926 (38,94%)
URLs of institutional services (LMS)	8,901 (50,04%)
Unique URLs pointing outside the institution after filtering	1,018

On average, the system recorded slightly under 150 URLs per user. The initial filtering step significantly reduced this number as more than one third of them were pointing to course material, and the number increases to half of them when considering other sites within the university (student email account, virtual folders, etc.)

The content retrieval step where the URLs are used to effectively fetch the document from the net was performed over the 1,018 unique URLs produced by the filtering step. The course under consideration is part of a bilingual program (English/Spanish). As a consequence, the retrieved content is analyzed to detect the language and the documents are divided into two collections: English and Spanish. The lexical analysis phase was applied to a collection of 25 reference documents and 252 student documents in English, and 25 reference documents and 599 student documents in Spanish. The results reported in this paper are for the English documents.

The entire processing phase, from collecting the URLs until the similarity matrix is obtained took less than 15 minutes. Although a lexical analysis of a large set of documents could become a bottleneck, the previous stages of filtering and classification reduced the number of documents to a size manageable by current techniques within reasonable execution times.

As a result, the similarity matrix of size 25×252 is obtained. In order to select the most relevant documents from the student set, two different approaches were considered. The first was to calculate the accumulated similarity of a student document with respect to all the reference documents. This method will be referred as the ACC method. Alternatively, the documents were sorted in decreasing order of its maximum similarity to any reference document. This method will be referred as the MAX method. As a sample, Table 2 shows the ten URLs with the highest accumulated similarity (ACC).

When using the MAX method, the ten URLs with the maximum similarity to a single reference document had only two URLs in common with the set shown

in Table 2. The low number of URLs in common between these two methods suggested a validation strategy to try to characterize this difference.

Table 2. Ten resources with the highest accumulated similarity

N	Title	Host	Comment
1	Linked list	en.wikipedia.org	Page explaining what is a linked list. The course uses this structure in numerous activities.
2	Preprocessor directives	www.cplusplus.com	This compiler functionality is included in the course material and used throughout the course
3	Debugging with Gdb	sourceware.org	This tool is essential for the course and must be used frequently
4	Unicode	en.wikipedia.org	The course does not explicitly mention Unicode
5	Maemo Development Environment	maemo.org	Students develop an application that must execute in the Maemo platform.
6	Index of stdlib in C	www.thinkage.ca	The functions in this library are used in most activities
7	Linux essential keyboard shortcuts	linux.about.com	The development platform runs on Linux and the command line is used frequently
8	What is a pointer?	pw1.netcom.com/t̃jensen	Pointers are one of the most studied topics in the course
9	Apache Subversion	subversion.apache.org	Subversion is used to exchange documents between team members and instructors.
10	Linux Command Line Reference	www.pixelbeat.org	Useful to work in the Linux environment.

4.1 Statistical Validation

Automatically obtaining a set of documents that are similar to the course notes is not enough to validate the proposed approach. It remains to be proved that the selected documents are indeed relevant to the course. The adopted validation approach consisted on a survey given to the seven instructors with teaching duties in the course. A questionnaire was given with the URL of the 20 resources obtained with the ACC and MAX methods respectively. For each URL, the instructor was asked to review its content and answer the following two questions:

1. How is the resource related to the course? The answer was given in a five levels scale (5 = Very related, 4 = Somewhat related, 3 = Neutral, 2 = Not related, 1 = Totally unrelated).
2. If the resource is related (or somewhat related), this relation applies to 1 = the entire course, 2 = a subset of activities or topics, 3 = a single activity.

The first question is oriented toward proving that the set of automatically obtained URLs are relevant to the course (the algorithm works). The second question was posed to test if the two methods would offer different resources in terms of their scope of application. Intuitively, if a resource has a high accumulated similarity (selected by the ACC method), it is probably more adequate for the general course topic. Analogously, if a resource is selected by its high similarity to one single reference document, it is probably more related to the specifics of the activity in that document.

Thus, for each instructor, at most 40 data points were obtained (only those resources ruled "related" or "somewhat related" were considered in the second question. The first step in the validation was to obtain a single mark per instructor for the relevance of the two sets of URLs, so the average score was chosen. The obtained samples had a mean of $43,57$ and a standard deviation of $3,64$. Next, a null hypothesis test that the average relevance of the URLs in the ACC group is not larger than 40 (somewhat related) was performed The t-test returned a significance p-value of 0.02, thus the hypothesis can be rejected and the sample assumed to have a mean above the "somewhat relevant" level. The normal distribution of the sample was verified with the Shapiro-Wilk test ($W = 0.8969$, p-value $= 0.313$).

The analogous test was carried out for the URLs obtained with the MAX method. In this case, the normal distribution of the sample could not be verified, thus a Wilkoxon signed rank test was performed. In this case, the null hypothesis of the average not larger than 40 (somewhat relevant) could not be rejected (p-value $= 0.13$) suggesting that the relevance of these URLs is lower than those in ACC. Additionally, the hypothesis of the average to be not larger than 30 (neutral) can be rejected, confirming that this second set of resources have some relation with the course.

The second question in the survey was included to investigate if the two methods ACC and MAX discovered resources with a different granularity in its relevance. For each of the 20 resources, the average of the valuations given by the 7 instructors was obtained. The two samples (each with $n = 10$) had means 0.84 and 1.35 respectively, suggesting that method ACC discovers more generic URLs whereas MAX contains resources related to a single task. The two samples were compared with the Wilcoxon Signed-Rank Test. A significant p-value $= 0.01782$ was obtained concluding that the granularity valuations indeed come from two different populations.

As a summary of this validation, both discovery methods, ACC and MAX discover somewhat relevant resources, but in the case of ACC, they are conclusively relevant. Furthermore, the URLs discovered with ACC seem to be more relevant to the entire course, and those discovered with MAX typically apply to a more reduced set of course activities.

Once the validity of the automatic discovery procedure has been shown, the resources can be used as recommendations to the students during the course. Two approaches have been envisioned. First an unsupervised variant by which

the recommendations are automatically added to the course material. The second approach follows a "moderated recommendation scheme" by which instructors must approve a resource before being included as a recommendation for the students. Because the discovery procedure is implemented along the course, this second approach improves the confidence of teachers on the system, at the expense of increasing instructor load.

5 Conclusions and Future Work

Student's support on problem-solving processes is improved by real time recommendations. In order to increase the performance ratio and the accuracy of such recommendations, automation of document discovery and classification is required. In addition, the recommendation must follow an automated method, able to provide the best match out of the huge amount of retrieved data. In this paper, we provide a first-hand experience with students, which carry out a blended learning activity. During the problem solving activity, students surf the Internet and look for a number of resources. However, this activity provides too many visited documents, from which there is no clear evidence of usefulness to the activity as they can be both related as well as non-related to the course contents. Task monitoring and document retrieval become a challenge, then. The authors design, implement, and test a method to deliver an automatic selection process of resources, which will be suggested in the form of recommendations to the students. Specifically, the method is characterized by a step-by-step process, namely: user tracking, URLs retrieval, document comparison, document fitness to the course, and eventually, recommendation.

By using ANNIE (a set of algorithms for natural language processing), which is part of the Gate toolkit, the implemented method obtains an annotated view of each document, which feeds a matrix with coefficients that supports the ranking system for the documents, following an algorithm that provides specific vectors for every document, that show the similarity factor to the course reference material, and therefore, usefulness to the students.

Out of the empirical study carried out with 220 students over a 8-week period, and in order to select the most relevant resources, we use two discovery methods: The ACC method (calculate the accumulated similarity of a student document with respect to all the reference documents) and the MAX method (sort the document decreasingly out of its maximum similarity to any reference document). We conclude that both discover relevant resources to the course and are thus useful for document recommendation. However, ACC seems to increase the validity and fitness ratio to the entire course, while MAX is more efficient with a limited set of activities. Once a document is selected as relevant, it can be upgraded to a recommended one either by an automatic process, or a supervised process in which the teacher grants the selected document, at his workload expense.

In summary, this paper presents a supporting system for selecting the most relevant and useful resources among the ones browsed by the students beyond the

course materials. In order to maximize transparency and minimize interference and workload for students, a content-based approach is proposed for determining the relevance of the potential resources. Evaluation results from empirical data confirms the validity of the approach. First, relevance of the selected resources has been confirmed by teaching staff's validation, providing a sound base for recomendations based on the proposed method. Additionally, appropriateness of selected resources is further analysed depending on the granularity required; alternative methods are thus suggested for selecting the most appropriate resources when considering either global course recommendations or more specific ones for a certain task. Experimental evaluation validates also the performance of the system for its real time application during a course. Finally, based on the successful rates for relevance and opinions provided by the teaching staff, we are confident that the system earned their trust for using it in real classes.

Future research will work on the implementation of a recommender system that will be integrated in the learning environment delivered to the students (a virtual machine in this case). This recommender will be based on the algorithms discussed in this paper, providing potentially useful resources in real time during the course. Further evaluation is also planned, in particular considering students' point of view. This study will allow us to validate aspects like the quality of real time recommendations, and if students and course staff actually consider the recommender useful for a better development of the course. Finally, alternative strategies like collaborative filtering or hybrid approaches combining both content-based and collaborative filtering are also being considered.

Acknowledgments. The authors are truly indebted to Luis Sánchez Fernández for his insightful comments about the lexical analysis, as well as the teaching staff that participated on the experience. Work partially funded by the Learn3 project, "Plan Nacional de I+D+I TIN2008-05163/TSI", the Acción Integrada Ref. DE2009-0051, the "Emadrid: Investigación y desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" project (S2009/TIC-1650) and TELMA Project (Plan Avanza TSI-020110-2009-85).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Ashraf, B.: Teaching the Google-eyed YouTube generation. Education+ Training 51(5/6), 343–352 (2009)
3. Auinger, A., Ebner, M., Nedbal, D., Holzinger, A.: Mixing content and endless collaboration–MashUps: Towards future personal learning environments. Universal Access in Human-Computer Interaction. Applications and Services (2009)
4. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval, vol. 463. Addison Wesley/ACM Press (1999)
5. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and UserAdapted Interaction 12(4), 331–370 (2002)

6. Çelik, T.: Attention.xml Technology Overview (2005)
7. Cunningham, H.: GATE, a general architecture for text engineering. Computers and the Humanities 36(2), 223–254 (2002)
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, vol. 54, pp. 168–175. Association for Computational Linguistics (2002)
9. Drachsler, H., Hummel, H.G.K., Koper, R.: Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. International Journal of Learning Technology 3(4), 404–423 (2008)
10. Govaerts, S., Verbert, K., Klerkx, J., Duval, E.: Visualizing Activities for Self-reflection and Awareness. In: 9th Int. Conf. on Web-based Learning (2010)
11. Lops, P., Gemmis, M., Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends. In: Recommender Systems Handbook, pp. 73–105. Springer, US (2011)
12. Macfadyen, L.P., Dawson, S.: Mining LMS data to develop an "early warning system" for educators: A proof of concept. Computers & Education 54(2) (2010)
13. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender Systems in Technology Enhanced Learning, ch. 12, pp. 387–415. Springer, Heidelberg (2011)
14. Mazza, R., Dimitrova, V.: CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. International Journal of Human-Computer Studies 65(2), 125–139 (2007)
15. Mazza, R.: A graphical tool for monitoring the usage of modules in course management systems. In: Lévy, P.P., Le Grand, B., Poulet, F., Soto, M., Darago, L., Toubiana, L., Vibert, J.-F. (eds.) VIEW 2006. LNCS, vol. 4370, pp. 164–172. Springer, Heidelberg (2007)
16. Pardo, A., Delgado Kloos, C.: Stepping out of the box. Towards analytics outside the Learning Management System. In: Int. Conf. on Learning Analytics (2011)
17. Resnick, P., Varian, H.: Recommender systems. Communications of the ACM 40(3), 58 (1997)
18. Romero, C., Ventura, S., Garcia, E.: Data mining in course management systems: Moodle case study and tutorial. Computers & Education 51(1), 368–384 (2008)
19. Romero Zaldívar, V.A., Burgos, D., Pardo, A.: Meta-rule based Recommender Meta Systems for Educational Applications (2011)
20. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5), 513–523 (1988)
21. Salton, G., McGill, M.: Introduction to modern information retrieval, vol, vol. 1. McGraw-Hill, New York (1983)
22. Schmitz, H.-C., Scheffel, M., Friedrich, M., Jahn, M., Niemann, K., Wolpers, M.: CAMera for PLE. In: Cress, U., Dimitrova, V., Specht, M. (eds.) EC-TEL 2009. LNCS, vol. 5794, pp. 507–520. Springer, Heidelberg (2009)
23. Schmitz, H.C., Wolpers, M., Kirschenmann, U., Niemann, K.: Contextualized Attention Metadata, ch. 8. Cambridge University Press, Cambridge (2009)
24. Sifry, D., Marks, K., Çelik, T.: Attention.XML Draft Specification (2004)
25. Valsamidis, S., Kazanidis, I., Kontogiannis, S., Karakos, A.: Course Ranking and Automated Suggestions through Web Mining. In: 2010 10th IEEE International Conference on Advanced Learning Technologies, pp. 197–199. IEEE, Los Alamitos (2010)
26. Wolpers, M., Najjar, J., Verbert, K., Duval, E.: Tracking actual usage: the attention metadata approach. Technology Education & Society 10(3), 106–121 (2007)