Knowledge Computing Method for Enhancing the Effectiveness of a WWW Distance Education System

Alexandra Cristea and Toshio Okamoto

The Lab. of AI & Knowledge Computing
The Graduate School of Information Systems
The University of Electro-Communications
Chofu, Chofugaoka 1-5-1, Tokyo 182-8585, Japan
{alex,okamoto}@ai.is.uec.ac.jp

Abstract. The ultimate aim of our research is a free, evolutionary, Internet-based, agent-based, long-distance teaching environment for academic English. For this purpose, we are building 2 environments: student learning and teacher courseware design environment. Here we focus on the second research direction, on constructing the teacher authorware environment (courseware management), and especially on a new method of automatic, knowledge computing-based courseware indexing that uses the AI paradigm of Concept Mapping.

1 Introduction

We are building a system to help academicians learn to exchange meaningful information with their peers in English. The system is WWW hypermedia based. The detailed rationale of the research, the system modules and the student environment design were shown in [3]. The system is also a remote authoring system, for teachers designing the courseware. With the growth and spread of the Internet, and at the same time, the constantly increasing transmission bandwidth, many applications are moving towards the net. Benefits are, among others, the wider reach compared to stand-alone applications. For educational applications, another benefit is the extension of the time concept. WWW courses can be accessed at any time, from anywhere, not only when the teacher is in the classroom. However, recently, the rigidity of the existent Web course material has been noticed. The research community generally agrees that knowledge processing and adaptive indexing benefit the end-user. The end goal of this line of research is to design courses which offer the study material in the best way possible for the current student, adapt to his/her needs, knowledge, misunderstandings, representations, and so on. One type of adaptation is via automatic, adaptive courseware indexing. Courseware indexing has been studied by many researchers [2], who have discussed the benefits and dangers of automatic indexing. To decrease system complexity for teacher and student user, and offer correct automatic guidance, we propose indexing as a trade-off between complete automation, and complete teacher input.

P. Brusilovsky, O. Stock, C. Strapparava (Eds.): AH 2000, LNCS 1892, pp. 289-292, 2000.

2 Information, Organization, and Linking

The information exchange from tutor to system contains input of lessons, texts, links between them, etc., but also requests for help in editing. The data from the tutor is stored in 6 different structured databases, including a library of expressions that appear in the text, a Video-on-Demand (VOD) database, a background image database, an audio database of listening examples, a full text database and a link database.

A. Structuring Information into Knowledge

The information that the teacher inputs is restricted from start to small units of building blocks, called *texts*, that can have a complete audio or video version. These texts can be grouped in *lessons*. Each text/lesson has the following attributes: main text (for text units only), a short title, keywords, explanation, patterns to learn, conclusion, and finally, exercises. We established this structure, as a first step towards efficient indexing, so that we obtain an easy retrieval by a search of information from titles and keywords, as is natural, but also in explanation and conclusion files.

B. Test Points

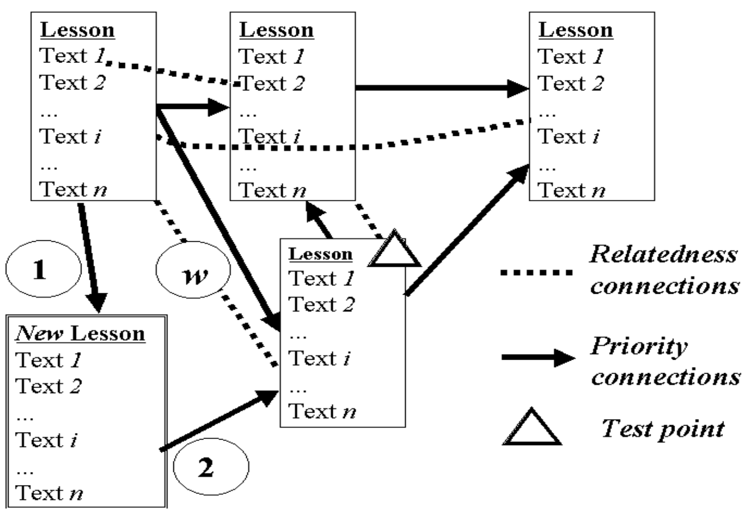

Fig. 1. The subject link database.

The teacher should mark *test points*, at which it is necessary to pass a test in order to proceed with the course. If the student wants to jump one or more subjects, s/he can proceed with only one test, made of a random combination of tests from the previous test points, in a proportional relation. In this way, the teacher can make sure that the student will not pass any level without having actually understood the subject. In this way, aimless searching through the html pages is avoided. Next to these obligatory tests, there can also a number of optional tests.

C. Priority and Relatedness Connections

When introducing one or more subjects (fig.1), the teacher has to specify the *Priority Connections* of the learning order, by building a directed graph (of pointed arrows). When there is no order, subjects will have the same priority, and build a set. Priorities among the texts of a lesson are set implicitly according to the order of the texts, but can be modified, if necessary. The teacher can also add connections between related subjects, with indirect links, called *Relatedness Connections*, for subjects, among which no specific learning order exists, but which are related. These relations are useful, e.g., during tests: if one of the subjects is considered known, the other one should be also tested. The main differences between the priority connections and the relatedness connections is that the first ones are directional, weightless connections, whereas the latter are non-directional, weighted connections.

D. Concept Mapping

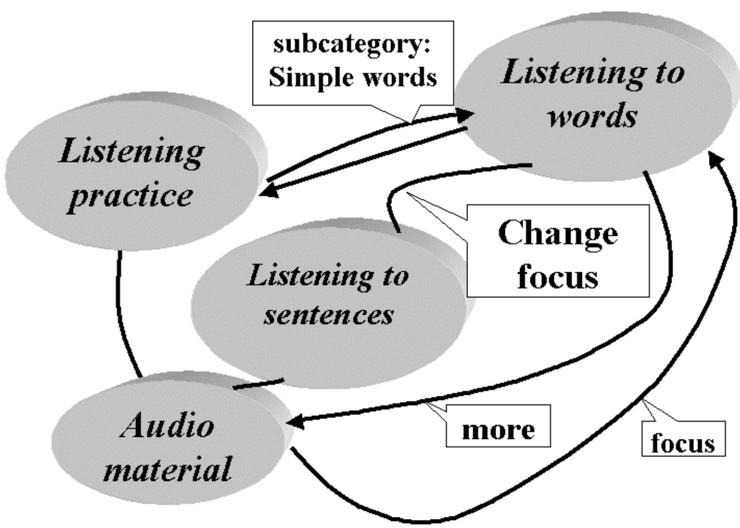

Fig. 2. Labeling example

Concept mapping [1,4] can be applied in two ways in order to improve the connectivity and information content of the subject linking. The teacher inputs more *transversal links* via concept mapping techniques: (1) The system offers a keyword list and a title list, and the teacher can select concepts, and link them in a graphical editing window. In this way, brainstorming and free generation of related concepts is supported. (2) The system *automatically links* concepts, then presents the map(s) to the teacher, for inspection. The teacher can add, delete links, and, more important, can label links. When labeling, the teacher can introduce also labeling directions, as shown in fig.2. The teacher is only responsible to link his/her own course to the existing courses created by him/her or some of the other courseware designers.

E. Subject Relatedness Weight Computation

The relatedness connections have weights that change interactively, reflecting how connected' two subjects are:

$w_{A,B}^{0}=\{$ **1:** teacher selection; **0.5:** system generation; **0:** rest;$\}$. **(1)**

$w_{A,B}^{t+tconst} = \alpha\, w_{A,B}^{t}$ + f1(no. of times connection $_{A,B}$ activated) + **(2)**
+ f2(no. of times connection $_{A,B}$ was accepted, when proposed in relation to unknown subject) +
+ f3(no. of times connection $_{A,B}$ was accepted, when proposed in relation to query) +
+ f4(no. of times tests related to connection $_{A,B}$ were solved satisfactorily or not); .

$w_{A,B}^{0}=\{$ **1:** teacher selection; **0.5:** system generation; **0:** rest;$\}$.

where: $\alpha\in(0,1)$: forgetting rate; f1~f4: linear functions; $w_{A,B}>0$: weight between subjects A and B; t: time; tconst: period for weight update.

3 Conclusion

We showed here in short the adaptive indexing and connection mechanisms of our evolutionary, Web-based, academic English teaching tool, MyEnglishTeacher". We focused on courseware subject connection optimization design, for easy, efficient retrieval of relevant course information, based on teacher input, and on Concept Mapping techniques for the subject-link generator. From system courseware design requirements, we enforce the generation of structured content databases, as a basis to knowledge bases for concept mapping. We showed the weight update function computation for the relatedness links. With the priority graph built by the teacher, the relatedness graph automatically built by the system, and the concept mapping based linking and labeling, student guidance in the multimedia web courseware is possible.

References

1. Beyerbach, B. (1988) Developing a technical vocabulary on teacher planning: preservice teachers' concept maps", Teaching and Teacher Education, Vol. 4.
2. Brusilovsky, P., Schwarz, E. and Weber, G. (1996) ELM-ART: An Intelligent Tutoring System on the World Wide Web", Intelligent Tutoring Systems, Eds.: C. Frasson, G. Gauthier & A.Lesgold, Lecture Notes in Computer Science, Vol. 1086, Springer, 261-269.
3. Cristea, A. and Okamoto, T. (2000) MyEnglishTeacher – An Evolutionary, Web-based, multi-Agent Environment for Academic English Teaching", Proc. of Congress on Evolutionary Computation, San Diego, USA (to appear).
4. Kommers, P.A.M. (1996) Conceptual Support by the New Media for Co-operative Learning in the Next Century", Multimedia, Hypermedia and Virtual Reality; Models, Systems and Applications, Eds.: P. Brusilovsky, P.A.M. Kommers and N. Streitz, Springer.