


to the coordinators of courses or units of learning. This work is motivated by the need of automating and facilitating the search of experts that could contribute to the design and development of units of learning in collaborative authoring tools. The proposed algorithm solves a set covering problem by following a greedy approach that maximizes a Gaussian cost function. The results show that the recommended collaborative teams are optimal in terms of precision, but should be improved in terms of coverage and recall.

Index Terms—Collaborative authoring, collaborative teams, recommendation algorithm, set-covering, coverage, precision, recall.

I. INTRODUCTION

A hot topic in e-learning is focused on developing authoring tools to support the creation of personalized learning design. Numerous tools exist providing support for building and execution units of learning. IMS LD is a specification to represent e-learning courses that supports the utilization of different pedagogical methods. Some known authoring tools are: ReCourse, Prolix-GLM, Collage, MOT, CopperAuthor, e-LD, etc. [1]

Some of them offer useful collaborative features. For instance, MOT 2.0 allows multiple authors to contribute in the authoring process, following the principle that the higher the number of users that contribute to the authoring process, the better the quality of the final material [2]. Thus, the authored materials foster a new level of knowledge about some specific material. Fig. 1 illustrates the problem. Set $E$ (circle) represents the set of topics belonging to any given course. In the same figure, set $A$ (big dotted set) describes the set of topics really mastered by the Coordinator or Main Teacher of the course. The remaining topics of the course (set $E'$) should also be covered and naturally opens the door for collaboration. Therefore, the problem consists on finding the minimum number of optimal collaborators for the uncovered set $E'$. A possible solution is illustrated in Fig. 1 with a collaborator represented by the set $B$ (small dotted set), which covers some keywords of the Coordinator of the course $(A)$ as well as keywords from the uncovered set $(E')$.

III. PROPOSED ALGORITHM

A. Goals

In what follows we describe the main goals we have to satisfy in order to search for optimal collaborative teams:

- **Goal 1.** To find a set of teachers that fully cover the keywords of the uncovered subset $(E')$, so that the coordinator of the course could have access to all the knowledge required to cover the set $E$.

- **Goal 2.** To find the minimum set of teachers satisfying Goal 1.

A basic solution to minimize the number of collaborators would consist on following a greedy approach by selecting those collaborators with maximum coverage of keywords of the uncovered set $E'$. However, we believe that the solution should satisfy some additional criteria, specially relevant for our problem. Two hypothesis are introduced to explain our line of thought:
Hypothesis I. A collaborator should have a certain degree of similarity with the coordinator of the course. Similarity is informally defined as the amount of keywords of the course (set $A$) covered by a collaborator (set $B$). It is our assumption that if a collaborator does not have some similarity with the coordinator of the course, it would be difficult to have an efficient communication between them. Therefore, we have to look for teachers optimizing the required degree of similarity.

Hypothesis II. A collaborator should present a high degree of specialization on those keywords matching the uncovered set $E'$. Specialization corresponds to the amount of keywords from the uncovered set $E'$ covered by a collaborator (set $B$). We believe that a collaborator who covers too many keywords of the uncovered set $E'$ is not an optimal collaborator, as it would probably imply not being expert on covered topics. As in the previous hypothesis, we should look for collaborators optimizing the required degree of specialization.

From these two hypotheses, the next two goals follow:

- **Goal 3.** To maximize the degree of similarity.
- **Goal 4.** To maximize the degree of specialization.

B. Mathematical description

Before we start the mathematical description of our problem, let’s denote some relevant variables:

- $E =$ set of keywords of a course,
- $A =$ set of keywords of the coordinator of the course,
- $B =$ set of keywords of a collaborator,
- $X =$ similarity,
- $Y =$ specialization,
- $S, C =$ subsets of $B$

The set-covering problem is an optimization problem that models many resource-selection problems [6]. An instance $(E, B)$ of the set-covering problem consists of a finite set $E$ and a family $B$ of subsets of $E$, such that every element of $E$ belongs to at least one subset in $B$:

$$E = \bigcup_{S \in B} S$$  \hspace{1cm} (1)

We say that a subset $S \in B$ covers its elements. The problem is to find a minimum-size subset $C \subseteq B$ whose members cover all of $E$:

$$E = \bigcup_{S \in C} S$$  \hspace{1cm} (2)

We say that any $C$ satisfying (2) covers $E$.

We are going to solve this problem using a greedy approximation algorithm, which works by picking, at each stage, the collaborator showing a set $S$ that maximizes a cost function that depends on the similarity and specialization values. Those values are obtained from the following equations:

$$Similarity(A, B) = \frac{A \cap B}{A}$$  \hspace{1cm} (3)

$$Specialization(E, A, B) = 1 - \frac{(E - A) \cap B}{(E - A)}$$  \hspace{1cm} (4)

In order to satisfy Goals 3 and 4, we have chosen a Gaussian function as a cost function whose center represents the optimal degree of similarity and specialization. Therefore, we have to maximize the following cost functions:

$$\max_X \text{Cost}(X) = e^{-\frac{(x - x_0)^2}{2 \times \sigma_x^2}}$$  \hspace{1cm} (5)

$$\max_Y \text{Cost}(Y) = e^{-\frac{(y - y_0)^2}{2 \times \sigma_y^2}}$$  \hspace{1cm} (6)

where $x_0, y_0$ are the center of the Gaussian blob, and $\sigma_x, \sigma_y$ are the width of the Gaussian blob.

We can combine equations (5) and (6) in order to come up with a single cost function. So finally, our greedy approach will be guided by the maximization of a two-dimensional Gaussian function:
Greedy approach based on maximizing a cost function. A Gaussian function is used as a cost function, whose value guides the greedy approach.

$$\max_{X,Y} \text{Cost}(X, Y) = e^{-\left(\frac{(x-x_0)^2}{2 \times \sigma_x^2} + \frac{(y-y_0)^2}{2 \times \sigma_y^2}\right)}$$ (7)

where \(x_0 = 0.5, y_0 = 0.25, \sigma_x = 0.5, \sigma_y = 0.5\).

Fig. 2 depicts our cost function in a three dimensional space.

C. Recommendation algorithm

In this section we describe the recommendation algorithm we have developed for solving our problem. For each collaborator, the following steps are carried out:

1) Computation of similarity using formula (3) for each collaborator.
2) Computation of specialization using formula (4) for each collaborator.
3) Calculating cost function with formula (7) for each collaborator.
4) Sorting out the cost function of each collaborator by descending order.
5) Picking the collaborator that covers the greatest number of remaining elements of uncovered subset \(E^r\).
6) Removing covered part from subset \(E^r\), i.e. \(E^r = B\).
7) Repeating steps from 2 to 6 until no elements remain in the subset \(E^r\) or until the list of selected collaborators finishes.

IV. Evaluation

A. Experimental setup

In order to evaluate the outcome of the proposed recommendation algorithm we carried out an experiment in the context of the Bachelor Degree of Computer Science at the University of Santiago de Compostela. Fig. 3 shows the questionnaire that were presented to the coordinators of courses and whose main purpose is to collect the relevant keywords of each course. When a coordinator of the course selects a course from the list as well as her name from the list of coordinators, the summary (or abstract) and the content of the course is shown from the database of courses. This information helps her to think about key terms of the course. After introduction of key terms, the coordinator of the course could select those key terms for which she wants to look for collaborators. Furthermore, she could specify some requirements to collaborator she is looking for. This is done by asking several questions supplied at the end of the questionnaire.

We have finally gathered information of 23 out of 50 available courses.

B. Results

The results of the recommendation algorithm are synthesized in Fig. 4 and Fig. 5. Fig. 4 illustrates a matrix showing the collaborative team recommended for a certain course. The first row of the matrix represents keywords of the course, whereas the first column represents the Main Teacher and the teachers of the collaborative team. The light green color means the existence of the keyword in the list of keywords of the collaborator, whereas white color means the absence of the keyword in the collaborators list of keywords. At the end of the matrix a summary of the coverage of keywords is given which defines how much of keywords of the course are covered and which keywords are left without a collaborator. The specific results for each course are illustrated in Fig. 5. We can notice that the algorithm has produced a recommen-
Collaborative teams recommended for a subject. A table with cost function of each selected collaborator is given.

Fig. 4.

Number of collaborators per course (A). Coverage of keywords requiring collaboration (B).

Fig. 5.

dation for 15 out of 23 courses. The recommended teams are usually small and only 3 out of 15 courses show more than 1 collaborator. In terms of coverage, 10 out of 15 courses either equal or overcomes the 50%.

To analyze the quality of recommended teachers we have analyzed each result in terms of Precision and Recall parameters, which are widely used in the literature [7]. We have randomly selected some teachers to ask about the relevancy of the recommended teachers and categorized the results in the following groups: total number of recommended teachers ($N_t$), total number of relevant teachers ($N_r$), relevant recommended teachers ($N_{rt}$), relevant non-recommended teachers ($N_{nt}$), and irrelevant recommended teachers ($N_{it}$). Based on these values, the definitions of Precision and Recall parameters follow:

- **Precision**: Ratio of relevant recommended teachers to total number of recommended teachers: $P = \frac{N_{rt}}{N_t}$
- **Recall**: Ratio of relevant recommended teachers to total number of relevant teachers. $P = \frac{N_{rt}}{N_r}$

Table I summarizes the results from some randomly selected teachers and courses. The algorithm provides very good results in terms of Precision (100%) but it works poorly in terms of Recall. Other relevant result is that the algorithm does not generate irrelevant recommendations.

<table>
<thead>
<tr>
<th>Course</th>
<th>$N_t$</th>
<th>$N_r$</th>
<th>$N_{rt}$</th>
<th>$N_{nt}$</th>
<th>$N_{it}$</th>
<th>$P$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1.0</td>
<td>0.25</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1.0</td>
<td>0.5</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1.0</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**V. DISCUSSION**

The proposed algorithm obtains very good results in terms of Precision (Table I), but is rather limited in terms of Coverage (Fig. 5 B) and Recall (Table I). We have identified three main issues to explain those limitations: (1) We have gathered an insufficient number of experts in our experiments. The algorithm need more experts to find optimal candidates. (2) We have to integrate additional sources of information. Teachers manage other sources of information about possible
teammates (research activity, etc.) in order to identify relevant collaborators. (3) Basic keyword matching process. It is hard to find coincidences in terms of basic keywords and sometimes there are no overlapped subsets to work with.

As for related work, the APODSLE Recommender service aims at finding people within a certain organization who have expertise related to any given learning goal [4]. However, the comparison with our algorithm is difficult as APODSLE do not generate teams as recommendations.

For future work we have to compare the current algorithm with others based on semantic similarity. Moreover, we should extend the current solution with educational ontologies in order to expand the list of keywords, identify the relationships among them, and use the new keywords and relationships to develop new ways to compute the similarity and specialization values. Finally, the recommendation algorithm will be integrated in WebLD2, our IMS-LD authoring tool [1], to extend its current collaborative features.

ACKNOWLEDGMENT

This work was supported by Xunta de Galicia under grant 09TUR001E, and Ministerio de Educación y Ciencia under grants TSI2007-65677C02-02, and TSI-020301-2009-9.

REFERENCES