Toward Reasoning-based Recommendation of Library Items – A Case Study on the e-Learning Domain

Stefano Ferilli [1], and Liza Loop [2]

[1] *University of Bari, Via E. Orabona 4, Bari, 70125, Italy*
[2] *LO*OP Center Inc., 16511 Watson Road, Guerneville, 95446, CA, USA*

Abstract

A primary function of libraries and librarians is to deliver relevant, interesting and useful items to the library users. With the enormous increase and continuous expansion of digital library content, it is not feasible for the librarian manually to carry out this service, and automated systems are needed to identify the candidate items to propose, possibly to be further checked or filtered by the librarian. Standard recommendation techniques studied in the Artificial Intelligence literature may be unsuitable, both due to the complex specificities of the library domain, and to the need to explain the recommendations. We propose an approach based on formal reasoning and on advanced matching assessment functions, and discuss in this paper the matching function specifically. We show a sample application to the e-learning domain, where the library stores the learning objects and the users are students. This is a piece of the larger and ambitious KEPLAIR project, currently under development, aimed at building an advanced Intelligent Tutoring System.

Keywords

Recommendation, Explainable Artificial Intelligence, Intelligent Tutoring Systems

1. Introduction

A primary function of libraries and librarians is to deliver relevant, interesting and useful items to the library users. With the enormous increase and continuous expansion of digital library content, it is not feasible for the librarian to be deeply acquainted with all the items in the repository, and thus the need arises for automated systems that can identify the items to propose, possibly to be further checked or filtered by the librarian.

Recommendation techniques have been widely studied in Artificial Intelligence (AI), mainly applied to e-commerce. However, the objectives and constraints of recommendation in e-commerce are very different, and less complex, than in Digital Libraries. Usually recall (i.e., retrieving all potentially interesting items, at the cost of possibly including irrelevant ones) is preferred over precision (retrieving only interesting items, at the cost of possibly missing some of them). An explanation of the recommendation is typically not required, because the user may just watch at the suggestions and choose which of them, if any, to purchase. Last but not least, there are many fewer and simpler facets to consider for the recommendation. In contrast, for professional uses, such as digital library content recommendation, proposing a few but highly relevant items is needed. The features of the user and of the items to be taken into account are much larger and less well-defined and tightly coupled with the 'syntactic' content of the items. Explaining the suggestions is fundamental, both for the user and for the librarian, to make sure that the recommendation is properly understood by them.

The latter issue is especially critical. It falls in the realm of eXplainable Artificial Intelligence, or XAI, a new branch of research focusing on the need to explain the decisions of AI systems to their

users. Whilst for some applications the user may use black-box AI systems, and safely adopt their decision because, if wrong, they do not imply significant cost or impact on their activities or on the involved people, some human activities are more critical and require thoughtful decisions that cannot be based on simple statistical or numerical models. These must involve human-level reasoning so that the users may decide whether or not to use them, and why.

A branch of AI aimed at reproducing human reasoning is Knowledge Representation and Reasoning, based on (typically First-Order) Logics representations and techniques. This setting usually requires a Logic to provide the formalism for representing knowledge about the given domain, a Calculus to provide inference mechanisms that produce new knowledge starting from the available one, and an Ontology to define the kind of entities that can be handled by the system along with their properties and relationships. In turn, the calculus may adopt a multistrategy setting, in which different kinds of inference cooperate in order to boost the inferential power of the system and tackle different needs and issues of the application domain. E.g., deduction produces new knowledge that was implicit in the available one, abduction hypothesizes sensible knowledge that was missing in the available one, argumentation resolves conflicting knowledge, etc. Thanks to its flexibility, this setting may also support the need for dealing with many more, and more 'conceptual', facets and features in digital library items recommendation.

This paper proposes a logic-based approach to recommendation of items in a digital library. It involves an ontology that acts as a schema for the description of items and users to be represented in the knowledge base, a formalism borrowed by Logic Programming, and an inference strategy based on associative reasoning. A very initial prototype was implemented, and applied to the application domain of e-Learning, where the aim is recommending learning objects (the library items) to learners (the users), possibly after evaluation of a teacher (the librarian). Specifically, this work is part of a wider research effort aimed at defining and building an Intelligent Tutoring System, named KEPLAIR, which pervasively exploits AI solutions to support all the activities and stakeholders involved in an educational setting.

The rest of the paper is organized as follows. The next section provides some background and discusses relevant related work. Then, Section 3 describes the KEPLAIR platform, and Section 4 introduces our proposal, with a very preliminary evaluation in the e-learning domain. Finally, Section 5 concludes the paper and outlines future work directions.

2. Background & Related Work

Educational Technology studies the digital tools and instruments that facilitate learning, along with their theory and practice. E-learning aims at using ICT to improve quality and fruition of education. By leveraging multimedia content and interactivity, it can improve engagement and comprehension by the learner. The on-line version of this setting has boosted the opportunities and widened the scope over traditional education, especially concerning access to contents independent of time and place. E-learning courses are based on Learning Objects (LOs for short), each concerning a limited and well-defined portion of their contents, and serving specific learning goals. They are delivered by software platforms called Learning Management Systems (or LMSs for short), that usually provide additional features, such as user statistics (used contents, time of use, etc.), social functions, etc. One of the earliest modern visions for using computers to enhance human learning was articulated in the 1960s by Engelbart, who developed a conceptual model for augmented intelligence [8,9] and introduced many elements of collaborative computing (including networks, shared workspaces, hypermedia, and video conferencing) that are nowadays standard. Applying this vision specifically to the individual learning process, Kay anticipated personal and mobile computing devices that dominate today's education [10]. However, whilst today lots of open educational resources such as MERLOT [12] and MIT's Open Courseware [11] provide free curated content, and enquiries through location-aware search engines instantly offer course details in one's subject area, their vision has not yet been realized. To fill this gap, a platform called KEPLAIR was proposed in [1].

One of the key functions to make learning processes more learner-centered is the recommendation of learning materials and activities. The field of recommender systems is well-established in AI [16]. Two kinds of approaches proposed for this purpose are collaborative filtering (related to the user

behavior) and content filtering (related to subject knowledge). The main focus in the literature has been on commercial applications, where the seller's interests have higher priority than the user's advantages. Recommendation in educational applications has followed their lead: e.g., 'learners who chose this course, also chose that course' or 'the average rating for this module was 4 out of 5'. However, neither of these approaches accounts for a learner's unique motivations, experience, learning path, or personal preferences. These limitations have been particularly obvious when considering 'lifelong learning', where formal and informal learning are intertwined and learners are responsible for directing their own learning [14]. To understand a learner's context, recommender systems would need to adopt a knowledge-based approach that takes into account additional information such as background knowledge, learner history and preferences [15].

Some works have specifically addressed the problem of document recommendation, but typically relying on standard or off-the-shelf recommendation approaches [22,24,25]. Some have investigated other tasks that may help recommendation, such as document classification [23] or explicit tagging [26] or cross-language settings [25]. Others have focused on specific contexts [21]. Conversely, we propose a new general recommendation strategy, based on a different and wider set of features and on a novel similarity computation; also, differently from standard approaches based only on distances, our technique also considers the paths in a knowledge base that connect the items to the user and can be useful to provide explanations of the recommendations.

3. KEPLAIR

KEPLAIR (Knowledge-based Environment for Personalized Learning using an Artificial Intelligence Recommender) [1,18] is a project aimed at developing an on-line e-learning platform that pervasively exploits AI solutions to support all the activities and stakeholders involved in an educational setting. In this section we will provide an overview of the parts of its functionality, architecture and ontology more related to the personalized use of its document repository.

3.1. KEPLAIR's Functionality

Concerning the learners, KEPLAIR's mission is to support their autonomy, supplying optimal materials, tools, and contexts for accomplishing their goals. A major feature of KEPLAIR is its strong emphasis on personalization (also called *individualization* or *adaptive learning*), a need that has been long recognized by theorists and learning designers [2,3]. Considering a learner's stated goal, KEPLAIR will devise tailored subgoals to create learning paths, recommending suitable learning opportunities, resources and materials (the LOs) within these paths to make their educational experiences engaging and productive, tailored to their personal interests, abilities, and contexts. We define *educational experience* as a combination of learning objects chosen according to a learner's profile presented in an accessible environment in order to help a learner achieve his goals.

The LOs to be recommended can be harvested from many different sources including school and college teacher submissions, other KEPLAIR users, online repositories of educational resources, industry and government resources, internet search engine results, and others (e.g., Wikipedia). Even other people (school fellows, teachers or coaches, etc.) may be considered as 'LOs' by KEPLAIR. In order to foster socialization and social group formation, KEPLAIR will include a social network devoted to discussions and social exchange. In this way KEPLAIR will be able to aggregate virtual and physical communities of users based on shared profiles, recommending interpersonal connections for those learners wishing to participate in face-to-face interactions.

Any Internet search engine can return hundreds such LOs, but the learner must sift through these, and might lack the knowledge or experience to distinguish credible resources from those that are less trustworthy, or might not recognize or be able to construct a meaningful progression of materials and activities to lead to successful learning, or be overwhelmed by the quantity of materials. KEPLAIR's recommendations will be filtered for compatibility with each learner's specific physical and digital contexts as well as their personal preferences. KEPLAIR might also present information that challenges these preferences, in order to expand their understanding and interests rather than narrowing them [5,6].

3.2. KEPLAIR's Architecture

KEPLAIR is based on a layered architecture, as shown in Figure 1. The core layer consists of 3 main components. The Learning Manager simulates human interactions via AI, acting as a tutor, counselor, and personalized assistant to build personalized learning paths for users, recommend appropriate LOs, and build tests to check the performance of the learners. The Social Manager handles interactions among users and communities of users in the educational context. All users' interactions with these modules are recorded in a Log repository, also used by both modules to drive their behavior. KEPLAIR may use Learning Objects (LOs) stored in its own repository or located elsewhere on the Internet. The 'external' LOs are identified and collected by a dedicated module, the Harvesting Manager, that also stores and continuously updates metadata describing the LOs in a LO Metadata repository. These metadata are used and updated by the Learning Manager during its activity, also based on the users' feedback on KEPLAIR's recommendations.
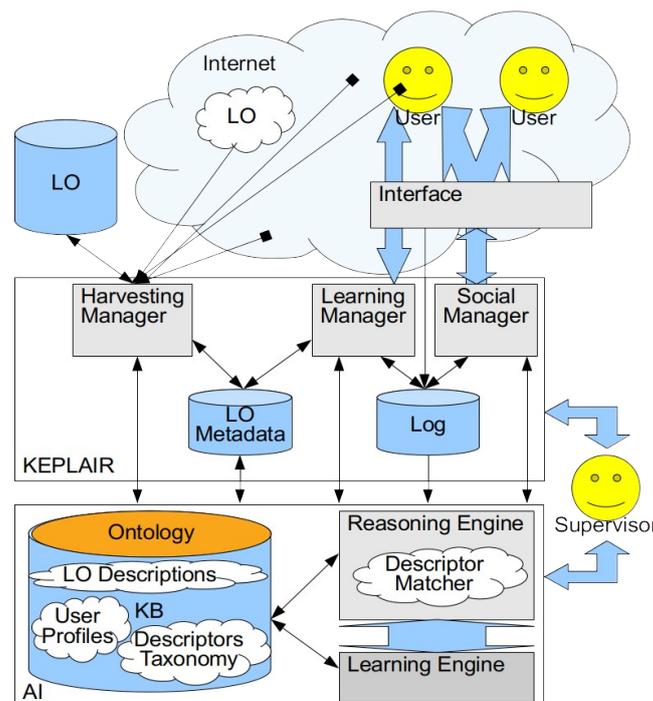
Figure 1: KEPLAIR's logical architecture

The core layer pervasively relies on an underlying AI layer implementing an intelligent agent according to the architecture proposed in [7]. Among other functions, it builds and maintains the user's profiles and the LOs' metadata on which recommendation is based, and includes the reasoning facilities to match such profiles for recommendation purposes. The inference engine integrates several different kinds of reasoning to be applied on its Knowledge Base (KB), on the LO Metadata repository, and on the Log. KEPLAIR places a strong emphasis on glass-box, symbolic AI techniques, so that its decisions and actions are explainable in human-level terms. Thus, human supervisors may check the behavior of the system, and can identify and fix wrong or biased decisions.

KEPLAIR's KB includes multi-faceted descriptions of the users (recording their background, skills, preferences, biases or limitations, trustworthiness, etc. [4]) and of the LOs (using different kinds of metadata to describe the resources and their use, including language, topic, complexity, correctness, etc.). Such descriptions are built and maintained (refined, updated) based both on information explicitly provided by the users and on information automatically extracted by the system from the Internet and the Log, namely:

- explicit feedback provided by the users (learners, trusted teachers and instructional designers) through diagnostic tests, questionnaires, rating functions, etc.;
- analytics of a growing base of anonymized learning data;

- implicit feedback obtained by analyzing the interactions of the users in the social section or with the LOs (e.g., keystroke or highlighting).

As the system accumulates experiences, feedback and human corrective intervention, its success rate in making recommendations that users find helpful will improve.

3.3. KEPLAIR's Ontology

As said, user profiles and LO metadata are stored in KEPLAIR's KB along with other kinds of knowledge. The schema that determines what the KB can store and how it does this takes the form of an ontology. In fact, the whole behavior of KEPLAIR is informed by such an ontology. At large, it deals with four kinds of information:

- **Goals.** Something an individual learner is curious about or wants to achieve. KEPLAIR will recommend a learning experience keyed to the goals and designed to help learners achieve them.
- **Profile.** A fully elaborated learner profile contains demographic information, a complete educational transcript, a résumé of experiences, relevant hobbies, cognitive strengths and weaknesses and, importantly, personal preferences. KEPLAIR uses users' profiles to filter all the possible learning experiences it discovers and to order their presentation to learners.
- **Learning Objects.** Humans learn by interacting with something or someone: all of these elements would qualify as learning objects. KEPLAIR will recommend learning objects that are at the student's level (transcripts), are presented according to their preferences for learning materials (cognitive profile) and are appealing to them (personal preferences).
- **Environment.** Humans are always immersed in a context or environment (social, physical, geographical and emotional, possibly digital). KEPLAIR takes into account the learner's environmental situation when selecting the learning objects to recommend.
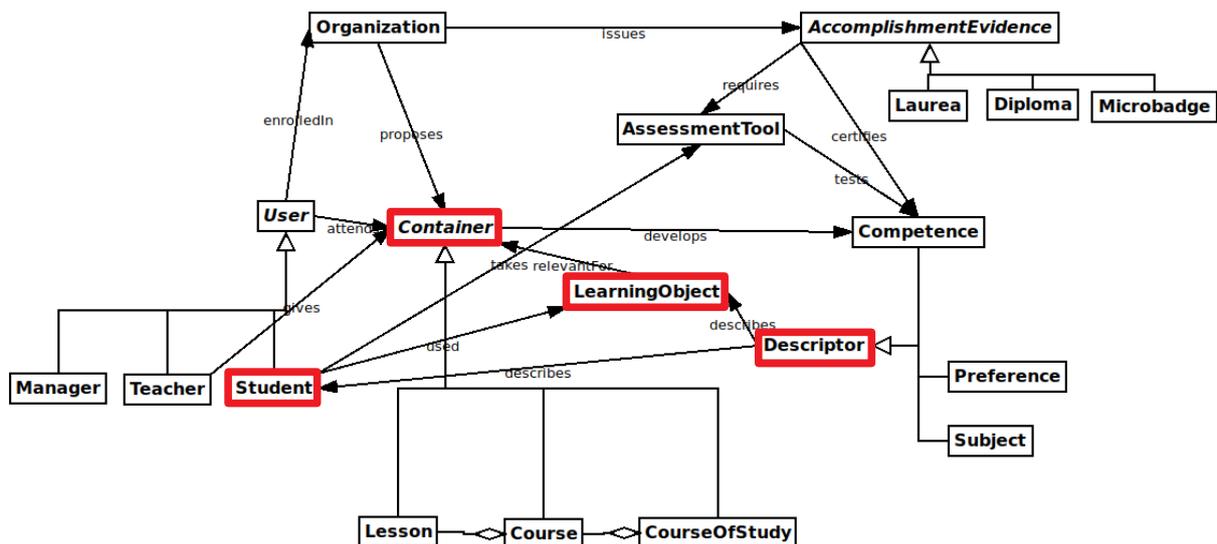

Figure 2: Portion of KEPLAIR's ontology. In red the classes used in the case study (see Section 4.2)

These areas are tightly related to each other, which results in a highly connected ontology. Figure 2 shows a sample portion of KEPLAIR's ontology. Entities are shown as boxes. The main entities are Organization, Container, LearningObject, AccomplishmentEvidence, AssessmentTool, Descriptor. Many of them have subclasses, e.g.: Manager, Teacher and Student are subclasses of User; a Container may be a Lesson, a Course or a CourseOfStudy; Laurea, Diploma and Microbadge are types of AccomplishmentEvidence; etc. Especially relevant to our purposes is the Descriptor class. Its instances will be used to describe user profiles and LOs. Many different facets of these descriptions are expressed by subclasses; in the figure we see Competence, Preference and Subject. In addition to the specialization relationship between classes and their subclasses, the figure also shows meronimy

relationships Lesson.partOf.Course and Course.partOf.CourseOfStudy. Other domain-specific relationships, shown as arrows, say that Organizations issue AccomplishmentEvidences, that AccomplishmentEvidences certify Competences and require AssessmentTools, that AssessmentTools test Competences, that Users are enrolled in Organizations, etc.

4. Recommendation Strategy

In this section we propose the basics of our recommendation approach. Each user and library item is associated in the KB to a set of descriptors, explicitly assigned by its creator, by the different kinds of users, or automatically extracted from usage analysis, user interaction, inference or other Data Mining approaches. Descriptors may express many kinds of information about the library items: obviously their subject, but also the mode or media of presentation, possible requirements or biases, etc. In the same way, they may describe many different facets of users: skills, preferences, etc. Explaining in detail how these descriptors are associated to users and Documents is out of the scope of this paper; here we will assume that they are already available in the KB. In this section we will adopt the DL perspective, and thus will refer to Documents instead of LearningObjects.

4.1.   Recommendation Approach: Method and Application

We propose a 2-step reasoning-based recommendation approach, that should ensure the required flexibility and also foster explainability of recommendations:
1. *Candidate selection*: uses automated reasoning to obtain a set of Documents to be recommended.
2. *Descriptor expansion*: uses automated reasoning to identify additional descriptors for the selected Documents and for the User (other than those explicitly associated to them in the KB).
3. *Candidate ranking*: uses automated reasoning plus set and mathematical operations to assign a degree of matching between two Documents, or a User and a Document, based on their sets of descriptors.

The role of reasoning is crucial, since it can establish non-trivial connections between users, Documents and Descriptors even when they are not explicitly linked in the KB. Many kinds of inference strategies can be used and combined. At this early stage of our research, we propose the use of associative reasoning, based on following the links expressed by relationship instances in the KB. Associative reasoning is not knowledge-aware. It just follows the links that connect objects in the KB, independently of the kind of link. The only additional information it may use is about links that cannot be followed.

4.1.1. Candidate Selection & Descriptor Association

In step 1, selection may be used to find Documents relevant to a User, or related to each other, even if the user or the other Documents have no explicit link to them. These objects might be the initial set of candidates for recommendation. E.g., a Document might be a candidate for recommendation to a User due to the following path present in the KB:
    User.used.Device.builtBy.Person.workedFor.Company.citedIn.Document
In step 2, 'Subject' Descriptors might be associated to Users or Documents due to the following paths in the KB:
    User.used.Document.relevantTo.Subject
(the set of descriptors associated with the Documents used by a User are relevant to describe the User)
    Document.citedIn.Document.relevantTo.Subject
(the Descriptors associated with Documents cited by Documents used by a User are also relevant to describe the User)
    Examples of links that cannot be exploited in associative reasoning are on 'Language' Descriptors:
    Italian.instanceOf.LatinLanguage.hasInstance.Spanish
(because if a User can speak Italian it does not mean he can also speak Spanish to some degree).

4.1.2. Matching Degree Assessment

In step 3, the candidate Documents selected in step 1, described by the Descriptors explicitly associated to them or identified by automated reasoning, must be ranked by some kind of matching degree to the reference User or Document (again, described by the Descriptors explicitly associated to them or identified by automated reasoning). So, we need a way to compute the matching degree based on two sets of descriptors.

Since the set of descriptors provided by the system may vary over time, adding instances to the Descriptor class, the use of feature vectors to describe users and library items, and of all matching techniques based on such a representation, are not viable. Instead, we should turn to techniques based on set operations on descriptors. Moreover, descriptors are not isolated pieces of information. They are often organized into taxonomies (e.g., 'Article' and 'Book' are both specific kinds of the 'Printable' type of 'Medium') and connected by several kinds of relationships (e.g., skills in 'Mathematics' may be needed to understand a book on 'Boolean Algebra'). If these taxonomies and relationships are stored in the KB, we should leverage them in our matching degree computation, going beyond simple set operations on descriptors. Additional challenges come from the fact that different types of objects to be matched (e.g., users to library items), might be described by different types of attributes. So, techniques based on simple set operations, such as Jaccard's index[2] [19,20], are insufficient, and we need more advanced ways of identifying and handling the sets of descriptors to be associated to users and library items. One proposal that can deal with taxonomic relationships was proposed in [17]. However, here we also want to take into account other kinds of relationships.

Inspired by Jaccard's index, we propose to compute a distance based on the following strategy:
1. in each set of descriptors, remove the descriptors that are above others in the descriptors' taxonomy;
2. consider their set product, i.e., all pairs of descriptors (d',d'') where d' is taken from the former set and d'' from the latter set;
3. for each pair (d',d'') compute the number of hops in the path connecting d' to d'' in the descriptors' taxonomy;
4. compute the ratio of the sum of defined distances in the pairs over the total number of pairs.

The distance between two descriptors equals 0 if the two descriptors are the same; it is undefined if there is no path in the taxonomy connecting the two descriptors. In the following we will denote the latter case by symbol '−'.
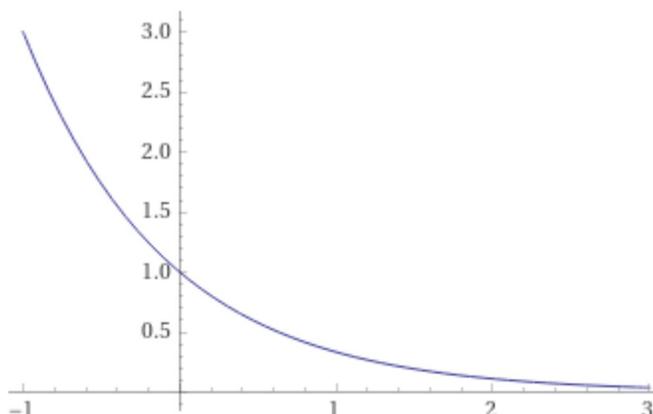
Figure 3: Plot of the score function for comparing two sets of descriptors

Call x the distance value computed in the last step of the above procedure. The score function must be inversely proportional to the distance. We define it using the formula $s(x) = 1/a^x$ with a > 0. Since

---
[2] Jaccard's index of similarity between two sets X and Y is given by:

$$J(X,Y)=\frac{|X \cap Y|}{|X \cup Y|},$$

which evaluates to 1 for two sets that are equal, or to 0 if they have no intersection.

the distance can never be negative, the score can never be greater than 1. The higher the distance value, the lower the score computed by this function. More specifically (see Figure 3):

- if $x = 0$, then $s(x) = 1$ (it equals 1 when the two sets have distance 0, i.e. they are the same);
- for $x \to +\infty$ $s(x) \to 0$ (it goes asyntotically to 0).

This behavior is independent of the choice for parameter a (it only determines the curve slope).

We further smooth the result of our score function by multiplying it by a penalty value that takes into account the unrelated descriptors, $P = 1 - r^b$, where r is the ratio between the number of pairs of descriptors with undefined distance over the overall number of pairs, while b is a parameter affecting the weight of the penalty. If all the tags are related to each other, then $r = 0$ and no penalization will be applied to the score (it will be multiplied by 1).

4.2. Case Study on e-learning

Our recommendation approach is general-purpose and applicable to any digital library. Since in this paper we are discussing its specific application to e-learning in the KEPLAIR project, in the following we will show a running example taken from this domain. In general, the mappings between the generic and domain-specific application are:

- Digital Library = KEPLAIR repository of LOs
- Library items = LOs
- Library user = Student

4.2.1. Example of computation

For our purposes, we focused on the portion of ontology enclosed in the red circle in Figure 2, including classes Student, Container, LearningObject and Descriptor. In particular, we considered the LearningObjects used by Students, and the Courses for which those LearningObjects are relevant. By describing the portion of KB associated to a given Student, and specifically the LearningObjects he already used, our aim is recommending new LearningObjects that he has not used yet, based on:

- the descriptors directly associated to users;
- the descriptors directly associated to LOs;
- the descriptors associated to LOs used by the students.

For the sake of demonstration, in Figure 4 we show a tiny portion of descriptors taxonomy we used in our experiment. Red lines denote paths that are forbidden in the computation of distance. So, e.g., English and Italian are unrelated (it does not make sense to say that someone who knows English may somehow also be interested in Italian documents, unless we know he also knows Italian). E.g., based on these relationships the distance between descriptors 'Theory' and 'Physics' is 1, between 'Manual' and 'Programmable' it is 3, while between 'People' and 'Active' it is undefined (–).

In this example, and in our experiments, let us use parameters a = 3 for the score function, and b = 0.5, i.e. the square root, for the penalty. Then, a sample computation with all the intermediate steps unfolded is the following.

Input tagsets for user (U) and item (I)	U = {players, people, electronic, active} I = {people, theory, physics, electric}
Reduced tagsets	UR = {people, electronic, active} IR = {people, physics, electric}
Pairs	{people-people, electronic-people, active-people, people-physics, electronic-physics, active-physics, people-electric, electronic-electric, active-electric}
Distances between pairs	D = [0, –, –, –, –, –, –, 2, 3]
distance using our proposed formula	DJ = (0+2+3)/3 = 5/3

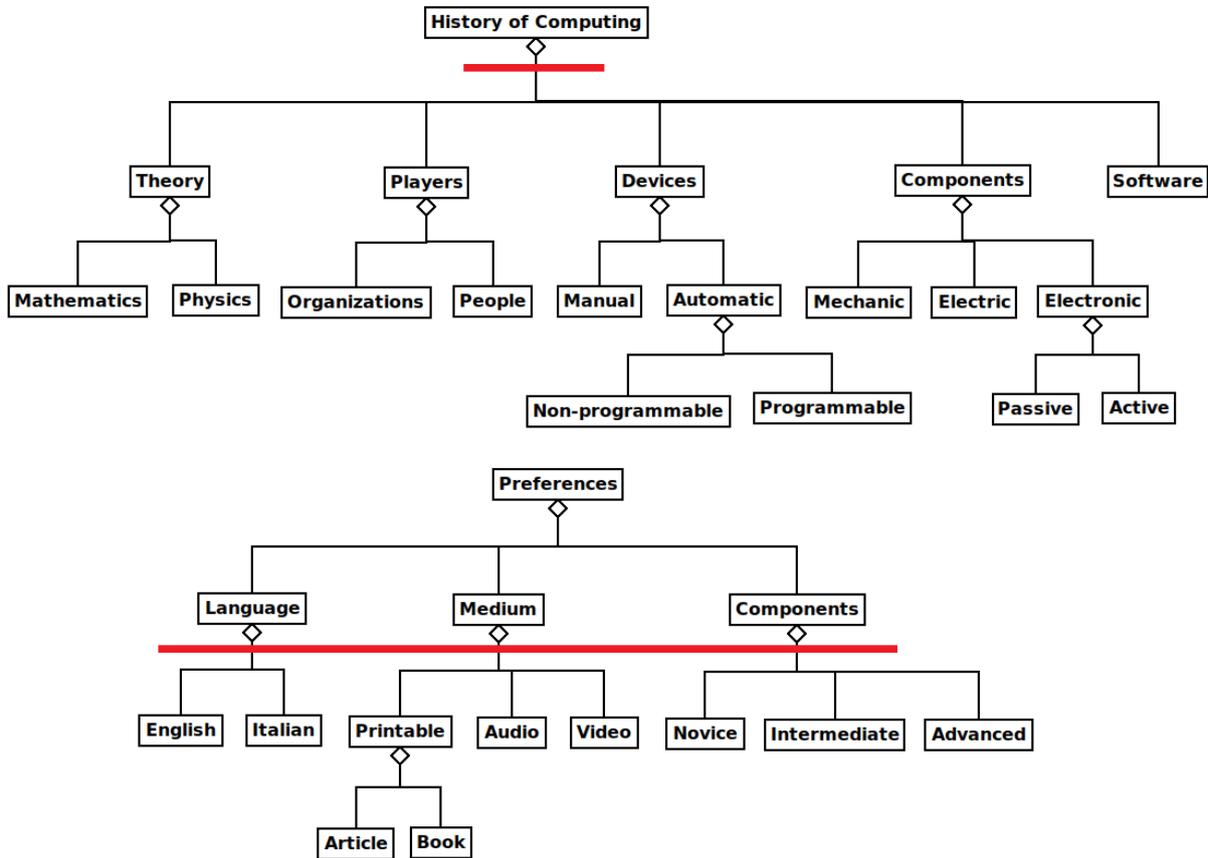| score function	$s(5/3) = 1/3^{5/3} = 0.1602$
penalty	$P = 1 - \sqrt{(6/9)} = 1 - \sqrt{0.667} = 1 - 0.8165 = 0.1835$
penalized score	$S = 0.1602 * 0.1835 = 0.0294$

Figure 4: Fragment of descriptors taxonomy used in our experiment

4.2.2. Prototype Implementation

We implemented our approach for a preliminary evaluation.

The KB was implemented using GraphBRAIN [13], a technology and platform for KB management and exploitation in the form of Knowledge Graphs. It brings together the data handling power and efficiency of Graph DB technology with the flexibility and reasoning power of ontologies. In GraphBRAIN, ontologies act as DB schemas to determine what can be stored in the DB and how, and as a connector to plug high-level automated inference into the knowledge. A prototype of the GraphBRAIN platform is already available and running, and currently used to store knowledge about cultural heritage in general and, specifically, about the history of computing. It provides for the Document class useful to store the records of a digital library, currently including nearly 4000 class and relationship instances specific to the computation domain, plus other classes aimed at describing domain-specific knowledge (about devices, components, software, systems, and configurations) and contextual information (persons, companies, events, places, subjects). Part of the information was automatically extracted from text documents using the procedure described in [27]. It also stores information about the users of this knowledge, and several kinds of relationships among instances of these classes. So, we used it as the KB for a hypothetical course on the history of computing.

As the e-learning platform in which to embed the proposed recommending strategy we chose the OpenOLAT LMS software (https://www.openolat.com/), written in Java. Since OpenOLAT does not provide for learning object tagging, we added this feature in the "Description" section of the LOs.

Tags must not contain spaces and start with a hash character (#). In the prototype, the platform proposes the LO recommendations in two different sections: the main pages, showing the list of all courses, and the specific pages of each lesson. In the former, LOs are recommended based on the past activity of the learner. In the latter, LOs that are close to the subject of the current page are recommended, neglecting the user preferences. This behavior resembles video recommendation in YouTube, where different strategies are used for recommendations in the homepage than for those shown beside a specific video in its page.

4.2.3. Experiment

To check the viability of our solution, we set up a brief experiment involving 5 people aged 30-60 and 50 documents (books) from the digital library of LOs stored in GraphBRAIN for the "history of computing" domain. The users were requested to select any number of descriptors taken from the ACM subject Computing Classification System (CCS) taxonomy (https://dl.acm.org/ccs), plus the descriptors concerning language and type of medium, as could best fit their interests and preferences. The users were also requested to list the 10 documents more interesting to them, ordered by decreasing interest, to be used as a gold standard. Using the same sources of descriptors, the documents were tagged with any number of descriptors useful to describe their contents. In the KB, the users and documents were associated to contextual objects (places, persons, organizations, etc.). So, we applied our associative reasoning-based strategy to select the candidate documents for each user and to associate descriptors to users and documents. More specifically, we used the following setting:
- taking all the documents within 5 hops from the user in the KB;
- taking all the descriptors within 2 hops from an object (user or document) in the KB;
- forbidding the links between different languages or media descriptors in the KB.

Then, the proposed formula was used to rank the candidate documents by matching degree with the user. If the reasoning produced more than 10 candidate documents, only the 10 candidates with highest score were returned. Then, the ranking produced by the recommender and that of the user were compared to check their relationship.

Table 1 reports for each user the number of candidate documents selected by the recommender, the number of such documents also present in the user's gold standard, the percentage of the recommended items that were also in the gold standard and the correlation index taking into account also the positions of the items in the recommendation and in the gold standard. Note that there is a chance that some documents not selected by the users are nevertheless useful to them. Our experiment adopted a cautious setting in this respect, considering as useful only the documents that were *a priori* selected by the users, not asking the users to check the usefulness of the suggestions *a posteriori*.

Table 1
Experimental results

User	#candidates	#hits	% hits	Rank Distance
A	14 ($\rightarrow$ 10)	7	70%	2.71
B	1	1	100%	4.00
C	5	3	60%	1.67
D	6	3	50%	3.33
E	5	4	80%	2.25
Average	6.2 ($\rightarrow$ 5.4)	3.6	72%	2.79

For user A, 14 candidate documents were retrieved by step 1 of our procedure, of which we considered only the 10 with highest matching degree for our statistics. User A is somehow an outlier, since for the other users no more than 6 candidates were found (6.2 on average, or 5.4 if considering only the 10 items actually recommended to user A). Recall that our candidate selection strongly depends on the contents of the KB, and thus some documents might not be considered as candidates just because they are beyond the 5-hop limit. The average number of hits on these recommendations is

3.6, corresponding to an average percentage of 72%. This is not bad, meaning that more than 2/3 of recommendations were able to guess the interests expressed *a priori* by the user. Also, the percentage of hits never falls below 50%, meaning that our technique seems to be effective also in each single case, not only on average. Only one recommendation was returned for user B, but it was correct. We have to consider that the users were required to choose 10 documents, but it may be that they were really interested only in the top items in their ranking, and that the bottom part of the ranking was filled in just to reach the required number of items. Checking in more detail the correspondence of ranking positions between the recommendations and the gold standard, the average distance is 2.79 positions, with a minimum of 1.67 and a maximum of 4. On a ranking of 10 items, this is also not bad. Of course, these results are not statistically significant. Still, they suggest that our approach is viable and that it is worth working on it to further refine and improve it.

5. Conclusions & Future Work

The enormous increase and continuous expansion of digital library content makes it infeasible for librarians to manually deliver relevant, interesting and useful items to all library users. Automated systems are needed to identify the candidate items to propose, and possibly submit them to the librarian for further checking or filtering before proposing them to the users. While there is a wide literature on recommendation techniques in Artificial Intelligence, standard approaches may be unsuitable for many reasons. First, they are typically oriented to commercial applications, where the seller's satisfaction prevails over the user's. Second, recommendation in the library domain may be much more complex, due to a much wider and open set of descriptors, and to direct and indirect interrelations among descriptors. Third, this is a crucial activity that may also require an explanation of the recommendations. To tackle all issues we propose an approach based on formal reasoning and on advanced matching assessment functions, and present in this paper the initial steps toward that goal. We applied our prototype in KEPLAIR, an advanced Intelligent Tutoring System, where the library stores the learning objects and the users are students. Results of a brief experiment, while not statistically significant, suggest that our approach is viable and worth further development.

Future work includes running more experiments on a wider dataset (users, documents and KB) and studying the performance under different experimental settings. Then, we would like to add other inference strategies in our candidate selection step, in order to more thoroughly exploit the content of the Knowledge Base and find more peculiar but less obvious candidate documents. Next, we should investigate the explainability in human-understandable terms of our recommendations, and the specific application to the e-learning domain and to KEPLAIR specifically.

6. References

[1] S. Ferilli, L. Loop, W. Rankin, P. Trafford, Introducing KEPLAIR – A Platform for Independent Learners, in: L.G. Chova, A.L. Martinez, I.C. Torres (Eds.), Proceedings of the 13th International Conference on Education and New Learning Technologies (EDULEARN 2021), IATED Academy, 2021, pp. 9638–9647. doi:10.21125/edulearn.2021.1943.

[2] J.R. Carbonell, AI in CAI: An Artificial-Intelligence approach to computer-assisted instruction, *IEEE Transactions on Human-Machine Systems*, 11(4):190–202, 1970.

[3] R.M. Ryan, E.L. Deci, Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions, *Contemporary Educational Psychology*. 2020 Apr;61:101860.

[4] P. Ocheja, B. Flanagan, H. Ueda, H. Ogata, Managing lifelong learning records through blockchain, *RPTEL*. 2019 Dec;14(1):4.

[5] Muller D., *Designing Effective Multimedia for Physics Education,* PhD Thesis, University of Sydney, 2008. Accessed 11 May, 2021. Retrieved from https://www.sydney.edu.au/science/physics/pdfs/research/super/PhD(Muller).pdf

[6] M. Kardas, E. O'Brien, Easier Seen Than Done: Merely Watching Others Perform Can Foster an Illusion of Skill Acquisition, *Psychological Science* 2018: 29(4): pp. 521–536. DOI: 10.1177/0956797617740646

[7] S. Ferilli, B. De Carolis & D. Redavid. An Intelligent Agent Architecture for Smart Environments. In: *Foundations of Intelligent Systems*. Lecture Notes in Artificial Intelligence 9384, 324-330, Springer, 2015.

[8] D.C. Engelbart, Augmenting Human Intellect: A Conceptual Framework, Fort Belvoir, VA: Defense Technical Information Center; 1962 Oct. Accessed 1 May, 2021. Retrieved from https://www.dougengelbart.org/content/view/138

[9] D.C. Engelbart, W.K. English, A research center for augmenting human intellect, in: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. New York, NY, USA: Association for Computing Machinery; 1968, p. 395–410. (AFIPS '68 (Fall, part I)). Accessed 6 May, 2021. Retrieved from https://doi.org/10.1145/1476589.1476645

[10] A.C. Kay, A Personal Computer for Children of All Ages, in: *Proceedings of the ACM annual conference* - Volume 1. New York, NY, USA: Association for Computing Machinery; 1972. doi: 10.1145/800193.1971922

[11] H. Abelson, The Creation of OpenCourseWare at MIT, *J Sci Educ Technol.*,17(2):164–74, 2008.

[12] T.E. Malloy, G.L. Hanley, MERLOT: A faculty-focused Web site of educational resources, *Behavior Research Methods, Instruments, & Computers*, 33(2):274–6, 2001.

[13] S. Ferilli, D. Redavid, An Ontology and Knowledge Graph Infrastructure for Digital Library Knowledge Representation, in: Digital Libraries: The Era of Big Data and Data Science, volume 1177 of Communications in Computer and Information Science (CCIS), pp. 47–61, Springer, 2020. doi:10.1007/978-3-030-39905-4_6

[14] H. Drachsler, H.G.K. Hummel, R. Koper, Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model, *International Journal of Learning Technology*, 3(4):404, 2008.

[15] J.K. Tarus, Z. Niu, G. Mustafa, Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning, *Artif Intell Rev.* 2018 Jun 1, 50(1):21–48.

[16] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems 46:109–132, 2013.

[17] S. Ferilli, M. Biba, N. Di Mauro, T.M.A. Basile, F. Esposito, Plugging Taxonomic Similarity in First-Order Logic Horn Clauses Comparison, in: R. Serra, R. Cucchiara (Eds.), *AI*IA 2009: Emerging Perspectives in Artificial Intelligence*, volume 5883 of Lecture Notes in Artificial Intelligence, pp. 131-140, Springer, 2009.

[18] S. Ferilli, D. Redavid, D. Di Pierro & L. Loop, Functionality and Architecture for a Platform for Independent Learners: KEPLAIR. In: Intelligent Systems Design and Applications – 21st International Conference on Intelligent Systems Design and Applications (ISDA 2021), Lecture Notes in Networks and Systems, 10 pp, Springer, 2022. (To appear)

[19] K.J. Horadam, M.A. Nyblom, Distances between sets based on set commonality, Discrete Applied Mathematics, 167, 2014, pp. 310–314.

[20] A. Gardner, J. Kanno, C.A. Duncan, R. Selmic, Measuring Distance between Unordered Sets of Different Sizes, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 137–143, doi: 10.1109/CVPR.2014.25.

[21] M. Borovic, M. Ojstersek: Document Recommendations in Slovenian Academic Digital Libraries. TPDL 2019: 361-364.

[22] F. Beierle, A. Aizawa, J. Beel: Exploring Choice Overload in Related-Article Recommendations in Digital Libraries. BIR@ECIR 2017: 51-61.

[23] A. Charalampous, P. Knoth: Classifying Document Types to Enhance Search and Recommendations in Digital Libraries. TPDL 2017: 181-192.

[24] Fuli Zhang: A Personalized Time-Sequence-Based Book Recommendation Algorithm for Digital Libraries. IEEE Access 4: 2714-2720, 2016.

[25] Y. Lai, J. Zeng: A cross-language personalized recommendation model in digital libraries. Electron. Libr. 31(3): 264-277, 2013.

[26] J. Alfredo Sánchez, Adriana Arzamendi-Pétriz, Omar Valdiviezo: Induced tagging: promoting resource discovery and recommendation in digital libraries. JCDL 2007: 396-397.

[27] F. Rotella, F. Leuzzi, & S. Ferilli. Learning and exploiting concept networks with ConNeKTion. *Appl Intell* **42,** 87–111, 2015. https://doi.org/10.1007/s10489-014-0543-z