International Journal on Digital Libraries manuscript No.
(will be inserted by the editor)

Ranking Dublin Core descriptor lists from user interactions

A case study with Dublin Core Terms using the Dendro platform

João Rocha da Silva[1], **Cristina Ribeiro**[1], **João Correia Lopes**[1]

INESC TEC, Faculdade de Engenharia da Universidade do Porto
e-mail: joaorosilva@gmail.com, e-mail: {mcr, jlopes}@fe.up.pt

Received: date / Revised version: date

Abstract Dublin Core schemas are the core metadata models of most repositories, and this includes recent repositories dedicated to datasets. DC descriptors are generic and are being adapted to the needs of different communities with the so-called Dublin Core Application Profiles that rely on the agreement within user communities, taking into account their evolving needs. In this paper, we propose an automated process to help curators and users discover the descriptors that better suit the needs of a specific research group in the task of describing and depositing datasets. Our approach is supported on Dendro, a prototype research data management platform, where an experimental method is used to rank and present DC Terms descriptors to the users based on their usage patterns. User interaction is recorded and used to score descriptors. In a controlled experiment, we gathered the interactions of two groups as they used Dendro to describe datasets from selected sources. One of the groups viewed descriptors according to the ranking, while the other had the same list of descriptors throughout the experiment. Preliminary results show that (1) some DC Terms are filled in more often than others, with different distribution in the two groups, (2) descriptors in higher ranks were increasingly accepted by users in detriment of manual selection, and (3) users were satisfied with the performance of the platform.

1 Introduction

Data is becoming an increasingly important research output. Studies prove that papers with associated data have higher citation rates, providing additional recognition of the authors' work [33,34]. Researchers need to share data, so informal data sharing takes place in spite of the lack of appropriate support for data curation [40]. There is a demand for datasets as a complement to research papers. On the other hand, high-quality metadata is vital for the discovery, retrieval and interpretation of research datasets [26].

The description of research data is recognized as an expensive process. While informal metadata is commonly created as researchers gather their datasets [21,37] and used to share data within research groups [31], the production of metadata good enough for sharing with the community requires much more effort and a higher degree of knowledge of metadata practices [7].

In an ideal scenario, researchers would work together with data curators to adequately capture the production context of a dataset [27]. In the current state of affairs, however, curators are unavailable in many research groups and there are few financial resources for data curation—especially on the so-called long-tail of science [18]. As a result, researchers often take the initiative to describe their data with publication in mind, and to comply with funding requirements [25].

The availability of user-friendly and comprehensive tools is an important motivator for describing and sharing datasets. Tools for research data management must balance flexibility, to address the needs of specific communities, interoperability, to favour exchange and aggregation, and ease of use, so that researchers can focus on the descriptions and let the tool take care of metadata representations behind the scenes.

Dendro is an open-source data management software platform aimed at the early stages of research data management [35]. It provides an environment where researchers can work collaboratively over *projects*, folder structures and files. Dendro can be configured to offer researchers a large choice of descriptors, plugged into the platform in the form of ontologies. But the more descriptors are available, the harder it is to select the right descriptors for a research group, user or domain. This is the rationale behind the idea of ranking descriptors

according to their usage and to the current project and user. In the proposed approach, Dendro uses past user behavior to help discover and use adequate descriptors for a dataset.

To test the ideas on descriptor ranking based on usage data, we have set up an environment where the interactions of users with Dendro are recorded. To limit the scope of descriptors, Dendro is configured to use only Dublin Core Metadata Terms Schema (DCTERMS) descriptors [10]. Dublin Core is the most widely used vocabulary in digital repositories, the meaning of the descriptors is well established and the most common ones are self-explanatory for users. Descriptors are selected and ordered depending on factors such as the resource being described or the previous use of descriptors by the users and their research groups.

The paper is organised as follows. Section 2 explains the Research Data Management (RDM) workflows based on Dendro, namely the principles adopted and the current results. Section 3 details the Dendro platform, with the main concepts and the supporting technologies. Section 4 describes the descriptor ranking approach, namely the selected features and the observations used to quantify them. The conditions of the user study are presented in Section 5. Section 6 discusses some analyses on the data collected in the study, followed by the conclusions.

2 Research data management workflows

The context of this work is the promotion of data management and data publication in long-tail research groups. We are working with several research groups from the University of Porto with the goal of setting up a complete workflow for managing their research data, right from the start of the projects, and comply with national and international data mandates [12].

There are many questions that arise when we meet the researchers, but almost all enquire about which metadata should be added to each file or set of files in preparation for deposit. To help solve this problem, our long-term goal is to design flexible metadata models that can:

- Incorporate descriptors from well-accepted vocabularies;
- Include domain-specific descriptors to enable reuse and foster research reproducibility;
- Be shared as linked open data to foster systems interoperability, allowing for the aggregation of datasets and improving discoverability.

We want to build these metadata models into research data management workflows that make the description tasks as easy as possible for researchers, while keeping their data and metadata easy to share and migrate. This is very important from a preservation point of view, and preservation is a central issue here. When

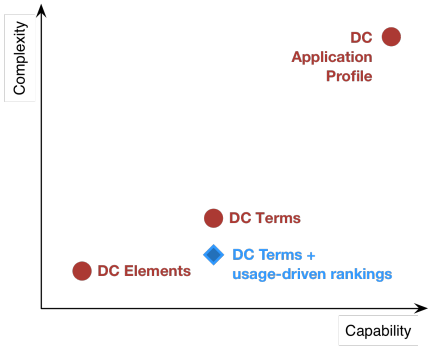

Figure 1: Tradeoffs in Dublin Core metadata

a researchers invests in preparing a dataset for publication, it is important to be able to make the metadata available as widely as possible, and to keep trusted metadata in case the institution or research group changes the storage and access system. This is in line with the FAIR principles, advocating Findable, Accessible, Interoperable and Reusable data and metadata [45], but also with the best practices in digital preservation [15].

2.1 Complexity vs. capability in DC metadata

The Dublin Core Metadata Element Set (DCMES) [11] is still widely used due to its simplicity, maturity and interoperability, even though some of its shortcomings were identified long time ago [1]. The DC elements were originally intended for lay users, and concepts such as the allowed range for a descriptor were not included. One of the results was ambiguity regarding the possible values for some descriptors. Questions such as *Is a string specifying the name of the creator of a resource valid as an instance of a* `dc:creator` *descriptor, or should it be the URI (Unique Resource Identifier) for the author?*[1] still arise in today's repository implementations.

The Dublin Core Terms specification (DCTERMS) was implemented in 2008 to deal with these issues and introduce finer, more detailed description semantics. The DCTERMS includes the original set of 15 descriptors specified in the DCMES as sub-properties of the original descriptors (to ensure compatibility with existing records), and adds several more, to a total of 55 [10]. The representation of DCTERMS in Resource Description Framework (RDF) enables the publication of metadata records as Linked Open Data [6], highlighting the importance of ontologies for the exchange of metadata records between systems [4]. While this extended schema can add much needed detail to the metadata records, it also increases the complexity of the data description task, as not every element is relevant when describing research results of different kinds.

[1] Example from `http://wiki.dublincore.org/index.php/FAQ/DC_and_DCTERMS_Namespaces`

Dublin Core Application Profiles (DCAP), a concept derived from that of an Application Profile [17], are the next step towards comprehensive metadata records. DCAP-compliant metadata records can include domain-specific and generic descriptors. The gathering and validation stages of the functional requirements include the final users in the process and an consider an adaptation period [9,30,1]. After the DCAP is in production it can also evolve according to the changes in data models of prominent repositories [23].

Figure 1 is an at-a-glance overview of the complexity of the three alternatives versus their ability to convey complex metadata. While DCMES is very easy to understand, it is quite limited in its ability to convey metadata appropriately. DCTERMS provides a much more detailed metadata model in RDF, but can be hard to understand by users without data management experience. Finally, DCAPs are very comprehensive metadata models but require prior knowledge of metadata practices and also of the domain they are intended for.

The descriptor ranking approach proposed here is implemented in Dendro to help users produce DCTERMS records for their metadata. The goal is to reduce the complexity of the task by gradually filtering the unnecessary descriptors while highlighting those that can be more relevant for each file or folder that the researchers are working on.

3 Dendro for data description

In the long tail of science, data management is mostly performed a posteriori—that is, at the end of the research workflow, and after publishing results. This places research data at risk, since researchers often seek new projects as funding ends, making it hard to obtain timely and comprehensive metadata for these large numbers of research products. To prevent this, the research data management process should start as early as possible, ideally as researchers have knowledge of their data production context and are actively producing their datasets [32,13,3]. This need has been identified in the past by projects such as ADMIRAL [19], and more recently by the EUDAT infrastructure, which provides separate solutions designed for the storage of data inside research groups (B2DROP) and for description and deposit into public repositories (B2SHARE) [24].

Dendro[2] brings dataset deposit and description closer to an earlier moment in the research workflow. It is a data management platform that acts as a file storage, description and sharing platform. It combines a "Dropbox"-like interface with some description features usually found in a semantic Wiki [36]. Users start by creating a project, which is a shared storage area among project contributors. Afterwards, they can upload files and create folders, while collaboratively producing metadata records for every element in the directory structure of the project [35].

Dendro has an extensible data model built on ontologies, allowing users to easily "mix-and-match" descriptors from different schemas in their metadata records. When compared to existing solutions supported on a relational model, this graph data model combined with full Linked Open Data representation allows the representation of metadata in a more flexible, interoperable and arguably, simpler way when compared to most repository platforms [38]. These alternatives often implement protocols such as OAI-PMH or specific REST APIs to expose their metadata records to external systems [2]. In Dendro we rely on URI de-referentiation—as proposed by the Linked Data guidelines [5]—and on SPARQL, which allows for much more sophisticated querying.

3.1 A brief overview of Dendro

Dendro allows users to create projects, which are very similar to the "shared folders" of Dropbox. Each user can see the projects where they can collaborate or that they have created. Project creators have the ability to invite other users to collaborate. After a user is added as a collaborator of the project, they will have the ability to upload files, create folders and edit metadata. Figure 2 shows a screenshot of the interface in charge of listing the projects that the current user has access to. The project creator will have access to access control features (**A**). The Administer function allows the creator to edit project-level metadata, while the Backup feature will export the entire project file structure to a BagIt [8] package; the ontology-based metadata records in Dendro are exported as RDF files and bundled into the package as well. These backups can also be restored back to Dendro should the need arise.

Figure 3 shows the view of a the root of a project in Dendro. It shows the file and folder browser (**A**) and the metadata record (**B**). Since the project metadata can only be edited in the administration area, the root folder of the project is always presented in the view-only layout, optimized for reading instead of editing metadata.

All project members can describe files or folder in the project's directory structure using the metadata editor, shown in Figure 4. The layout is divided into three main sections, from the left to the right: the file browser, which allows users to navigate in their files and folders; the metadata editor, which shows all the descriptors and controls for users to enter their corresponding values (text boxes, date pickers or maps, depending on the nature of the descriptor) and finally, on the rightmost position there is the descriptor selection area where users can pick descriptors to be included in the current metadata record. For each descriptor, two additional buttons are present: one to promote a descriptor to "Favorite", and

[2] Web site: `http://dendro.fe.up.pt/blog/index.php/dendro`
Source code: `http://github.com/feup-infolab/dendro`
Demo instance: `http://dendro.fe.up.pt/demo`

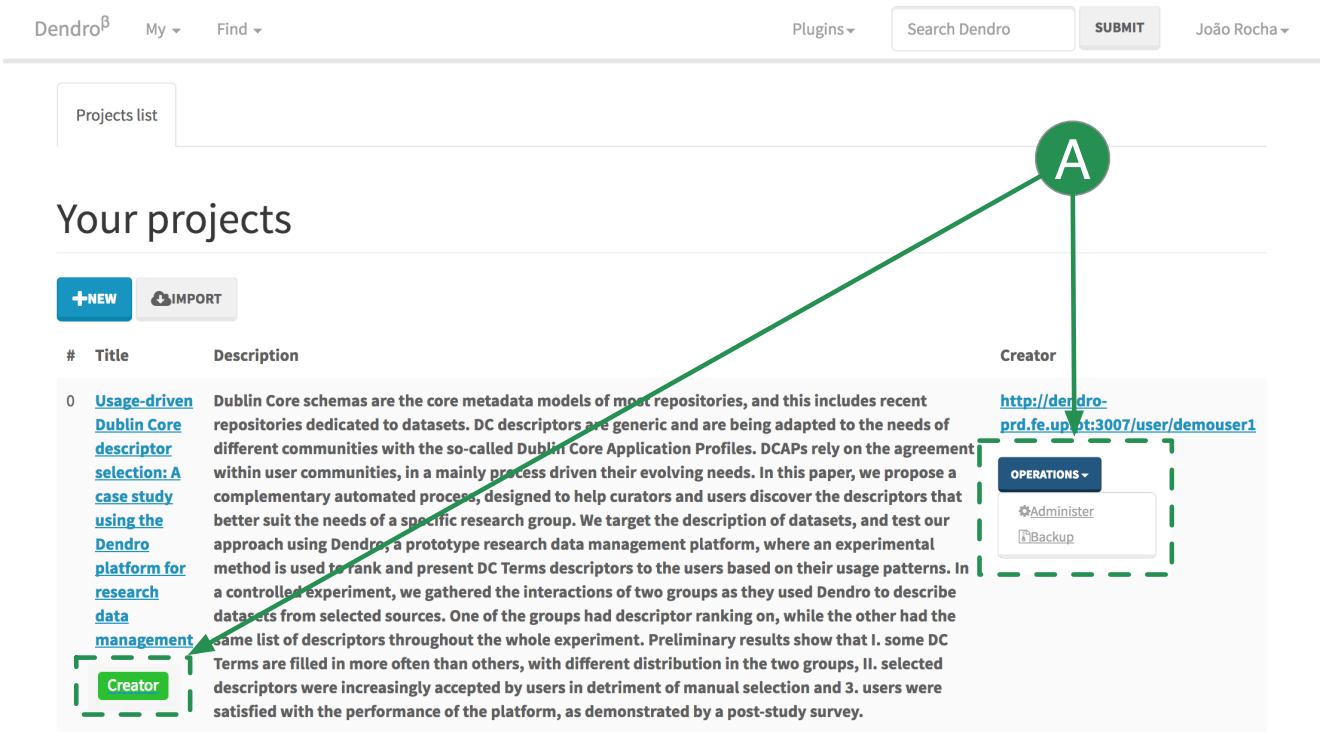

Figure 2: The list of projects in Dendro

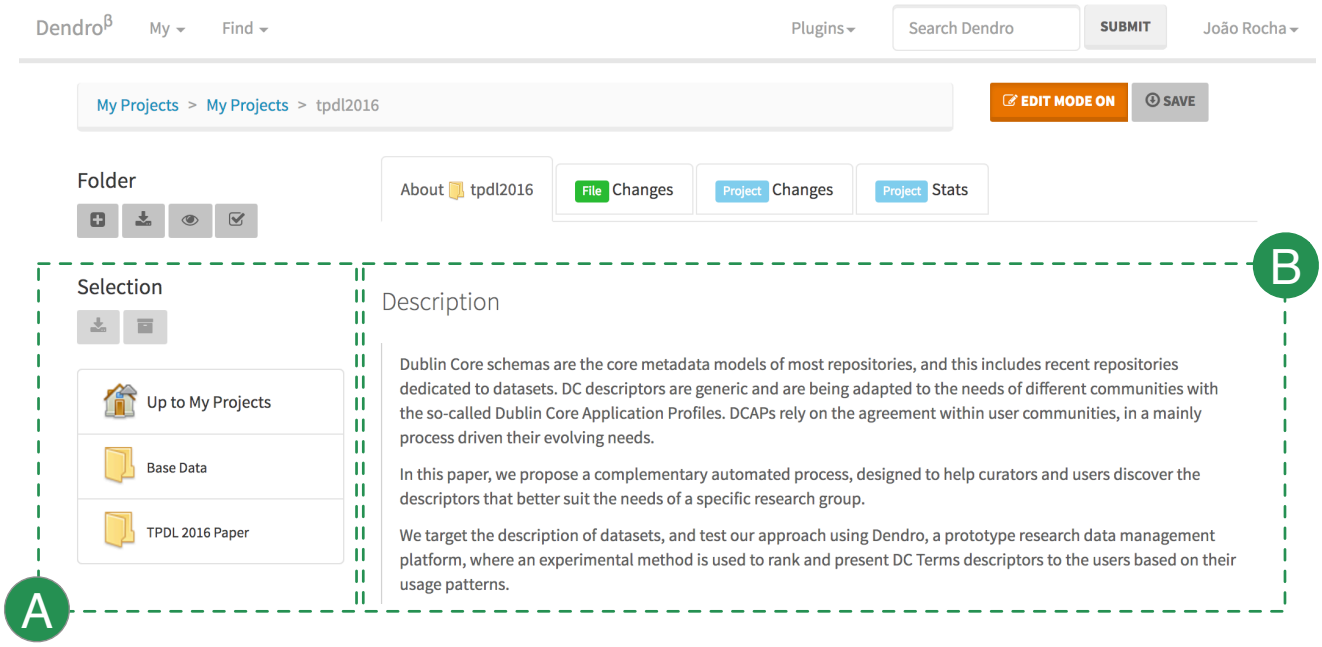

Figure 3: Viewing the root folder of a Dendro project

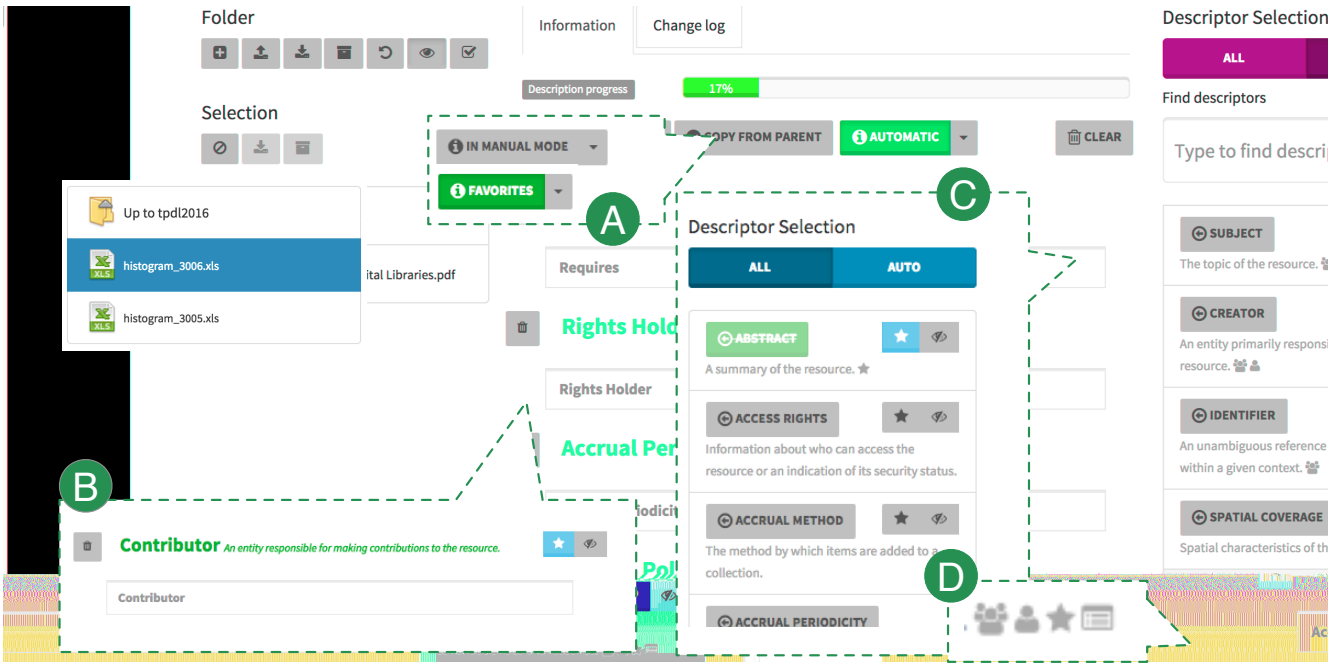

Figure 4: The main user interface of Dendro

another to "Hide" that descriptor. When the user selects the "Favorite" button for the first time, the descriptor will be marked as "Project Favorite" for all the project collaborators, moving it up in the list of suggested descriptors within that project. A second press will promote the descriptor to "User Favorite", making the descriptor go up in the list for the current user only, without influencing the lists presented to the other project collaborators.

A shows the interface modes that users can switch between. When the "Manual mode" is active, no descriptor suggestions are automatically added to the metadata editor; when the user switches to the "Automatic" mode, a set of descriptors selected by the platform are automatically added to the metadata editor at the centre, if they are not already filled in for the current record. In "Favorites" mode, descriptors that were marked as "Project Favorites" or "User Favorites" are automatically added. The interface highlights descriptors suggested via the "Automatic" mode in yellow and descriptors added via the "Favorites" mode in green (**B**).

Dendro also provides feedback to the user as to why descriptors are included in the list, in order to improve system *transparency* [42,44]. For each descriptor in (**C**), a set of icons can appear to indicate the reasons why a descriptor is included in the list (e.g. "Frequently used in project", "Frequently used in the entire platform", "Used in textually similar resources", etc.).

4 Descriptor ranking approach

Datasets require detailed metadata to make them usable by other researchers. Even comprehensive schemas such as DCTERMS may be insufficient, and a combination of generic and domain-specific descriptors may be required. However, as the number of available descriptors increases, the harder it is for researchers to determine which are relevant for their datasets. This is the main motivation behind the ranking of descriptors: bring to the attention of the researchers the descriptors they are likely to adopt. The data model of the Dendro platform is flexible due to its use of ontologies, and may be configured to combine generic and domain-specific descriptors in the metadata model. In the experiment where we test the descriptor ranking, however, only DCTERMS descriptors are used.

4.1 User interaction logs

To capture descriptor usage, Dendro was extended with several interaction logging capabilities. The use of the logs is twofold. On the one hand, logs are used to extract evidence of descriptor usage in several modalities. On the other hand, logs also contain user- and session-related information that can be used in experiments to evaluate the descriptor ranking approach. An excerpt of the log produced during a session is shown in Table 1. The first column (`uri`) is the identifier of the interaction. Dendro uses a graph data model, where every resource has an unique identifier, and interactions are treated as

resources. In this case, the identifier has the form of an URL, thus allowing the de-referentiation of the interactions by external systems, much like any resource in Dendro (e.g. `Files`, `Folders`, `Users`). The `timestamp` is the time when the interaction occurred and `user` is the username of the user who performed the interaction. The `type` of the interaction is the kind of action that the user performed—depending on this, it may be relevant to store a `ranking position`. Types have been identified according to the different actions that users may take in the course of a description session. A value of -1 in the `ranking position` means that the position is meaningless for the type of interaction in that entry. For example, when a descriptor is selected from the list on area **C** of Figure 4, we record its position in the list. Conversely, when a descriptor is selected after typing on the "auto-complete" search box above area **C**, there is no added benefit in saving its position on the list of "auto-complete" suggestions, since the search is typically directed to a specific descriptor. Users tend to type more characters until they can narrow down the first hit on the list, which they then select by pressing the Enter key. The "auto-complete" box filters descriptors by the `dcterms:title` or their `dcterms:comment` properties, allowing users to retrieve descriptors even if they do not know their exact name.

The types of interactions that are monitored in Dendro are shown in Table 2. Every interaction type has a name and a brief description. Each interaction is a piece of evidence of descriptor use or intention of use, and is used to define a quantitative evaluation of one or more of the descriptor features described below.

4.2 Descriptor features

A descriptor *feature* is a characteristic of a descriptor with potential to contribute to the rank of that descriptor in a list presented to the user. A set of descriptor features was identified (see Table 3), based on aspects of the descriptors that can be obtained from the user interactions. They were selected intuitively based on their capability to assess the degree of preference towards a descriptor. Aspects such as the descriptor that has been filled, the current user, the current project and the nature of the resource being described are relevant to the rank of a descriptor in the list.

The first features in Table 3, f_1 to f_4, can be regarded as capturing implicit feedback from the user. For example, filling in a descriptor is implicit feedback because it expresses an implicit acceptance of that descriptor without disrupting the description process. f_5 is a content-related feature, capturing textual similarity between the current resource and others in the system. Features f_6 to f_{11} represent explicit feedback from the user. Setting a descriptor as a favorite, for instance, is explicit feedback because it expresses a preference of the user with respect to the descriptor, and requires an action that is not a necessary part of the description work. Different types of feedback may contribute differently to the score of a descriptor, as they express different degrees of preference towards the descriptor.

4.3 Descriptor ranking

For each feature, we define a *component*, i.e. a numeric value that represents the contribution of the feature to the ranking score. The overall ranking of a descriptor is the sum of these components. Components are calculated from values obtained from the set of user interactions. In the long run, they will take into account the whole log of user interactions, or an abridged form of this information. The formulas proposed here for the components and for the descriptor scores are intentionally very simple, so that their effect on the lists of available descriptors can be fully understood.

To calculate a score value for a descriptor, we define the n *components* of the score for descriptor d, $c_{d,1\ldots n}$. For each feature, the component is the result of a formula that takes as input a measure associated with the feature (say the number of times descriptor d was used, for $c_{1,d}$) and generates the corresponding component value. The score for descriptor d is simply a sum, over all features, of the values of the n components, $c_{d,1\ldots n}$.

Adding the components produces the final score S_d for descriptor d.

$$S_d = \sum_{i=1}^{n} c_{d,i} \tag{1}$$

Table 4 details the components and the mathematical formulas that yield the value for each.

The formulas for the components are set empirically. The formulas for the components are similar for all features, except f_{10} and f_{11}. For feature f_i, there is a number of interactions that contribute to the component. They may be of a single type or accumulate interactions of different types. For example, accepting a descriptor within a project contributes to f_1 (frequent use of the descriptor) but also to f_3 (frequent user in the current project). An interaction count k_i is set for each feature, accumulating the relevant interactions. The formula for component $c_{i,d}$ is then obtained by multiplying an empirically defined weight that expresses the relative importance of this feature. In some cases, a maximum value for the component is also set; this is to avoid certain features from overwhelming others due to their weights. For feature f_1, for example, the component is

$$c_{1,d} = \min(+1.0 * k_1; +80.0) \tag{2}$$

The components for features f_2 to f_4 are similar. The maximum value in these components can make them top out quickly on their maximum values, compromising the ability of the ranking system to keep distinguishing between relevant and irrelevant descriptors in the long term.

Table 1: Excerpt of the interactions log

uri	timestamp	user	type	ranking position
http://dendro-prd.fe.up.pt:3005/user/201005026/interaction/2015-03-17T12:34:08.742Z	2015-03-17T12:34:08.742Z	http://dendro-prd.fe.up.pt:3005/user/201005026	accept_descriptor_from_quick_list	13
http://dendro-prd.fe.up.pt:3005/user/201104438/interaction/2015-03-17T12:35:17.200Z	2015-03-17T12:35:17.200Z	http://dendro-prd.fe.up.pt:3005/user/201104438	accept_descriptor_from_manual_list	49
http://dendro-prd.fe.up.pt:3005/user/201100868/interaction/2015-03-17T12:35:27.759Z	2015-03-17T12:35:27.759Z	http://dendro-prd.fe.up.pt:3005/user/201100868	accept_descriptor_from_manual_list	16
http://dendro-prd.fe.up.pt:3005/user/201005026/interaction/2015-03-17T12:36:13.539Z	2015-03-17T12:36:13.539Z	http://dendro-prd.fe.up.pt:3005/user/201005026	accept_descriptor_from_autocomplete	-1
http://dendro-prd.fe.up.pt:3005/user/201006772/interaction/2015-03-17T12:38:02.363Z	2015-03-17T12:38:02.363Z	http://dendro-prd.fe.up.pt:3005/user/201006772	accept_descriptor_from_manual_list	36
http://dendro-prd.fe.up.pt:3005/user/201005026/interaction/2015-03-17T12:38:53.404Z	2015-03-17T12:38:53.404Z	http://dendro-prd.fe.up.pt:3005/user/201005026	accept_descriptor_from_autocomplete	-1

Table 2: Types of interactions monitored in this experiment

Interaction type	Description
accept_descriptor_from_autocomplete	The user selects a descriptor from the autocomplete box in **B**.
accept_descriptor_from_manual_list	The user selects a descriptor from the list of descriptors by clicking on the corresponding button. The interface has to be in "All" mode, i.e. showing all descriptors ordered alphabetically.
accept_descriptor_from_quick_list	The user selects a descriptor from the list of descriptors by clicking on the corresponding button. The interface has to be in "Auto" mode, i.e. showing descriptors ordered by our ranking algorithm.
accept_favorite_descriptor_in_metadata_editor	The user fills in a descriptor automatically added to the metadata editor by the ranking system, while Dendro is in "Favorites" mode. When this mode is active, the current favorites (both for the user and for the project) are added to the metadata editor automatically on every page refresh. In this case, the user does not need to click the buttons on area **B** to add the descriptors to the editor in order to be filled in.
accept_smart_descriptor_in_metadata_editor	This is the same as the interaction type of the previous row, but the filled in descriptor is provided by the ranking algorithm. The top-15 descriptors are automatically added to the interface on every page refresh; if any of them was filled in, an interaction of this type would be recorded.
browse_to_next_page_in_descriptor_list	Clicking the "Next Page" button at the bottom of the descriptor list to see the next page of recommended descriptors. These interactions can indicate a poor performance of the system, because users will only click the button to move to the next page if they do not find what they want in the current page.
browse_to_previous_page_in_descriptor_list	Similar to the previous line. The user clicks the Previous Page at the top of the descriptor list.
favorite_descriptor_from_quick_list_for_project	Mark a descriptor as a favorite of the project. Useful when users want to "recommend" certain descriptors that other project collaborators should use in the descriptions. A project favorite descriptor will be a favorite, but only for the collaborators of the project and only while they are describing resources within its file structure.
favorite_descriptor_from_quick_list_for_user	Very similar to the previous interaction type, but this time the descriptor would be marked as a personal favorite of the user.
unfavorite_descriptor_from_quick_list_for_project	This will negate the "favorite_descriptor_from_quick_list_for_project" interaction. If there is, for a project, an interaction of this type registered at a later time than the last registered "favorite_descriptor_from_quick_list_for_project", the descriptor will not be presented as a favorite in the descriptor list ($\overline{\mathbf{B}}$) nor be automatically added to the editor when it is in "Favorites" mode.
unfavorite_descriptor_from_quick_list_for_user	Similar to the previous interaction type but for user favorites.

The component for f_5 is calculated from a similarity measure obtained from the search index used in Dendro (ElasticSearch). If there are many resources with similar text contents (only valid for certain file types from which text can be extracted), the descriptors used on those similar resources receive a score bonus. The component for f_6 through f_9 are calculated from explicit feedback over the suggestions. Setting a descriptor as a "Favorite" or rejecting it when it is suggested, for example, require the user to stop the description tasks to explicitly change descriptor status, so they have to be more strongly weighted than those in features f_1 through f_4. Finally, some descriptors will also not be presented, regardless of their score, if the user marks them as hidden for himself or for the current project (features f_{10} and $f11$).

The main reason behind the adoption of this simple ranking is that we can immediately observe changes in the ranking even after just a few interactions, helping to circumvent the cold-start problems usually found in recommender systems [39]. Other motivations are the limited timespan of the experiments with the ranking feature and the size of our user sample, which would

Table 3: Descriptor features

	Feature	Description
f_1	Frequent use	Is the descriptor filled in, regardless of user or project?
f_2	Recent use	Was the descriptor filled in by the current user in the entire Dendro instance, in the last 30 days?
f_3	Frequent use in the current project	Was the descriptor filled in, regardless of the user, but only in the same project as the file or folder being described?
f_4	Automatic acceptance	Was the descriptor filled in after Dendro automatically places it in the metadata editor? (Dendro in "Automatic" mode)
f_5	Textual similarity	Is the descriptor present in resources that are textually similar to the one being described? Valid only for files of certain types (`pdf`, `docx` and `txt`).
f_6	Rejection after automatic selection	Has the descriptor been removed manually from the metadata editor after it was added automatically by Dendro? (Dendro in "Automatic mode")
f_7	Acceptance of automatically selected favorite	The user has filled in this descriptor after it was automatically added to the metadata editor, in "Favorites" selection mode (see Area **A** of Figure 4)
f_8	Favorite for the current project	Is the descriptor a current favorite of the project?
f_9	User favorite	Is the descriptor a personal favorite of the user?
f_{10}	Project-level rejection	Has the descriptor been hidden by a collaborator of this project?
f_{11}	User-level rejection	Has the current user hidden this descriptor?

Table 4: Components of the ranking score for the descriptors

Component	Feature	Description	Formula
$c_{1,d}$	f_1	Descriptor d was filled in k_1 times over all records in the Dendro instance	$c_{1,d} = \min(+1.0 * k_1; +80.0)$
$c_{2,d}$	f_2	Descriptor d was filled in k_2 times by the user in the last 30 days	$c_{2,d} = \min(+2.0 * k_2; +80.0)$
$c_{3,d}$	f_3	Descriptor d was filled in k_3 times in the project that contains the resource being described	$c_{3,d} = \min(+2.0 * k_3; +80.0)$
$c_{4,d}$	f_4	The user has filled in descriptor d after it was automatically added to the metadata editor area	$c_{4,d} = +80.0$
$c_{5,d}$	f_5	Descriptor d is present in other k_5 Dendro resources that are considered textually similar to the one being described.	$c_{5,d} = \min(+20.0 * k_5; +80.0)$
$c_{6,d}$	f_6	The user has removed Descriptor d from the metadata editor after it was automatically added	$c_{6,d} = -80.0$
$c_{7,d}$	f_7	The user has filled in descriptor d after it was automatically added to the metadata editor, in "Favorites" selection mode (see Area **A** of Figure 4)	$c_{7,d} = +80.0$
$c_{8,d}$	f_8	Descriptor d was marked as a project favorite by the user or another collaborator of the current project	$c_{8,d} = +80.0$
$c_{9,d}$	f_9	Descriptor d was marked as a personal favorite by the user	$c_{9,d} = +80.0$
$c_{10,d}$	f_{10}	Descriptor d was hidden in the project that contains the resource being annotated	N/A (Forces hiding)
$c_{11,d}$	f_{11}	Descriptor d was hidden by the user, regardless of the active project	N/A (Forces hiding)

be insufficient for the application of conventional recommender algorithms such as collaborative filtering.

5 User study

To test our approach, we had the collaboration of 23 students of the Digital Archives and Libraries course at the Faculty of Engineering of the University of Porto (FEUP), 14 of which were women and 9 men. The median age was `24` and the average age was `29,2`, with a standard deviation of `9,87`. The course is part of the Masters of Information Science, so all students were already aware of concepts relevant for creating descriptions, such as metadata, metadata schema, descriptor or Dublin Core. The future career paths of these students may include curatorial roles such as librarian, digital repository manager, archive manager, or supervisor of any other system that requires information management expertise—in fact, some are already information science professionals. This is a small-scale study, in line with similar user studies in Information Retrieval and recommendation [22,43,14].

To run the experiment, two Dendro installations were set up, one with the basic configuration of alphabetically-ordered descriptors and the other with the descriptor ranking extension. Besides this difference, all features of the virtual machines and Dendro instances were exactly the same. Usage logs were collected in each Dendro instance.

We split our user group into two subgroups for A-B testing, to reduce learning bias i.e. allowing both groups to start the experiment without any prior knowledge of how to use the platform—thus allowing us to observe their learning behaviors separately as they interacted with the system. To improve the realism of this experiment, users were only instructed to provide the best possible descriptions for the datasets; the differences between the two versions of Dendro and the goal of evaluating descriptor usage were not discussed with them before the experiment.

The groups were named U_{Rec} (using Dendro with descriptor ranking) and U_{Alpha} (users of the Dendro with descriptors ordered alphabetically). Students were randomly attributed to U_{Rec} or U_{Alpha}. Each student was

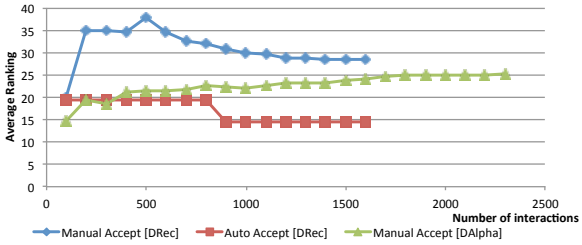

Figure 5: Average position of the selected descriptors in the selection lists

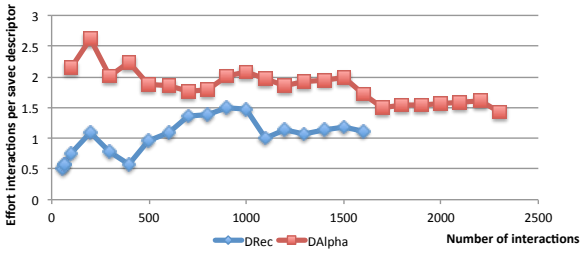

Figure 6: Effort interactions per saved descriptor

tasked, over three weeks, with the description of several datasets collected from different online sources and belonging to distinct research domains. Three sources of datasets were used: ICPSR[3], B2Share[4] and re3data[5]. In the case of re3data, which is a repository aggregator, the task included selecting a specific repository. Students were requested to go to the sources, select some datasets for which they could understand the contents, and use Dendro to add metadata and organize the associated files.

For each source, students created a project in Dendro and were allowed to collaborate, in pairs or groups of 3, provided they were all from either U_{Rec} or U_{Alpha}. This way we simulated a situation in real life were people collaborate in projects and discuss the most convenient metadata to assign. Nevertheless, each participant had their own tasks, in equal number for all.

While using Dendro, the interface elements in areas **A**, **B** and **D** (see Figure 4) were only available for participants in U_{Rec}. The **C** elements was available to all, but the "Auto" button was not enabled for users in U_{Alpha}.

6 Results analysis

The analysis of the logs resulting from the user study has to consider the interactions of the users in several tasks. We define *effort interactions* as those interactions that users have to perform to describe a dataset, in addition to filling in the descriptors themselves. An example of an effort interaction is adding a descriptor from the list **C** to the metadata editor **B**, by clicking on the corresponding button (see Figure 4). Conversely, those interactions that do not require any effort towards the descriptor lists are *non-effort interactions*. For example, Figure 4 shows several descriptors in yellow (which means automatically added) in the metadata editor at the center. Since the user did not have to manually select them, we record a non-effort interaction for every one of those descriptors, in case it is saved.

[3] https://www.icpsr.umich.edu/icpsrweb/
[4] https://b2share.eudat.eu/
[5] http://www.re3data.org/

Figure 5 shows a comparison of the average position of the descriptors as they were selected from the list. Lower values indicate an average position higher up in the list; the lower the value the better the ranking performs, since users find what they need in the top positions and do not need to look further down. For every value on the x axis (an interaction that occurred at an instant T), the corresponding y is the result of an average of the positions of the selected descriptors of all interactions that occurred before T.

From this chart we can see that the average ranking of the manually accepted descriptors in D_{Alpha} tends to 25. Since there are 55 descriptors in DCTERMS and D_{Alpha} shows the descriptors in alphabetical order, it seems natural that after a sufficient number of interactions the average position is near the middle position of the list. The "Manual Accept [DRec]" series shows the average position of the descriptors selected in D_{Rec} while it is in manual mode ("ALL" option is selected in **C**, see Figure 4). In this mode, D_{Rec} shows the same alphabetically-ordered list as present in D_{Alpha} so the long-term behavior is understandably similar.

The chart also shows the "Auto Accept [DRec]" series, which is the average position of the selected descriptors when the "AUTO" option is selected in D_{Rec} (see Figure 4, **C**). When this option is active, the order of the descriptors is given by the ranking algorithm. Since the average ranking of the selected descriptors is almost always lower than in the other two series, we can conclude that in this mode users consistently selected descriptors higher up on the list when compared to the alphabetical order. This is a positive outcome that indicates that the ranking improved the workflow of the users, since they found the descriptors they needed more easily when the option was active.

Note that not all interactions result in users filling in a descriptor—for example, users might select them from the list and then not fill them in. Since our most important measure of success is the number of descriptors that are actually filled in, we decided to plot the number of effort interactions for every descriptor that is filled in. Figure 6 shows the evolution of these ratios for the two Dendro instances. In the x axis are the effort interactions x_i recorded in the system, and the y axis values are calculated by dividing the number of effort interac-

tions recorded at the time of an interaction x_i by the number of descriptors present in the metadata records at that moment.

The D_{Rec} version shows a lower number of interactions for each descriptor that is saved, at all times. In the case of D_{Alpha}, this number is consistently higher than 1; since it has no automatic descriptor selection capabilities, users have to manually pick every descriptor (an effort interaction) from the list before they can fill it in. In contrast, the average number of interactions per descriptor in D_{Rec} is sometimes lower than 1. This happens because users fill in descriptors that are automatically added to the metadata editor instead of rejecting them (which is also an effort interaction), hence lowering the number of effort interactions required to fill in the same number of descriptors. This is another indication of an improvement introduced by the ranking approach versus the alphabetical ordering.

All the charts show a lower number of interactions overall registered in D_{Rec} when compared to D_{Alpha}—we believe this to be a consequence of a difference in performance and attention to detail between the user groups, although they were randomly split. We can also observe some learning behavior in our users. The first interactions in the system originate large variations in the number of interactions carried out per saved descriptor. As the number of interactions grows, the values stabilize and become lower, indicating an increase in efficiency by the users.

Figure 7 shows the distribution of descriptor instances by their DCTERMS element at the end of the user study. When comparing the two charts, it is apparent that the D_{Alpha} distribution has a larger "tail" when compared to D_{Rec}. Moreover, there is a higher frequency of descriptor usage for the top-n descriptors in D_{Rec} than in D_{Alpha}. This concentration of descriptor usage in the top-n descriptors can be explained by the way that the lists are produced, because they can highly benefit descriptors that have been somehow favored in the past.

This behavior can be positive, as it seems that users are liking the descriptor lists and that they are actually contributing to automate repetitive work. However, we may not exclude the possibility that this is due to other factors, such as the similarity between the datasets themselves—which we tried to counteract by requiring users to retrieve data from three different sources—or simply the inability of our algorithm to introduce previously unused descriptors in the lists.

6.1 User satisfaction survey

To complement our quantitative analysis, we performed an user satisfaction survey in line with similar work [43]. Our user survey was anonymous; we got answers from 18 users out of the initial 23 participants, 9 from each Dendro instance. 12 women and 6 men, with a median age of 23 years, average age of 27,7 and standard deviation of 9,63. Several user interface and user experience parameters were evaluated by the survey respondents, who graded each parameter according to a 1("Poor")-5("Excellent") scale. Following similar surveys [20,41], we outlined a series of questions specific to our case, which covered interface aspects—such as "Page Layout" or "Use of color". We reused these categories, but adopted a 1-5 star scale through which users rated each aspect.

The results of the survey are shown in Figure 8; the dot represents $\overline{p}$, the average rating for parameter p, while the error bars span across $[\overline{p}-\sigma_p;\overline{p}+\sigma_p]$, where σ_p is the standard deviation of the ratings for p. We observe almost identical user satisfaction marks for both Dendro instances. Users were more satisfied with the look and feel of D_{Rec}, and even more satisfied with its use of colour, as most answers reside in the $[3-4]$ bracket and the average is close to 4 versus an average score of 3.5. As seen in Figure 4, D_{Rec} uses more color elements to signal hidden and favorite descriptors and to highlight the reasons why descriptors are included in the lists—the positive response to the colours indicates that this information is useful to users.

The "Graphics" category shows similar averages in both instances, despite slightly lower scores in the D_{Rec} instance—this may indicate that the icons used should be improved. In the "Navigation" category there is a high variability of ratings in D_{Rec} when compared to the D_{Alpha} instance. There are more elements present in the D_{Rec} interface, so this result can be attributed to the additional learning that is required to make sense of the additional features. Since there are two distinct descriptor lists, users may become confused as to where they should go to select a descriptor. The "Page layout" category got very distinct results perhaps due to the additional scrolling that the users of D_{Alpha} had to perform to select their descriptors.

The "Usage instructions" category yielded very similar results for both instances, but with slight advantage towards the D_{Rec} instance, perhaps due to the additional functionalities that were present, which were always accompanied by user aiding elements such as pop-up tooltips. The "Descriptor display and explanations" parameter relates to the presence of short texts that provided explanations about the meaning of each descriptor. In this parameter D_{Rec} got worse results, but we could not pinpoint the exact cause, as both instances provide similar descriptor explanations in the same locations.

The results of the "Features" parameter were good, with an average score around 4 across both instances, with a slight advantage to the D_{Alpha} instance. "Reliability" and "Responsiveness" got overall positive scores, but there is certainly room for improvement. Some minor bugs were reported during the testing of the platform, and those may have negatively influenced the users' impression of the system. D_{Rec} performed worse in these categories, as more operations and more code was run

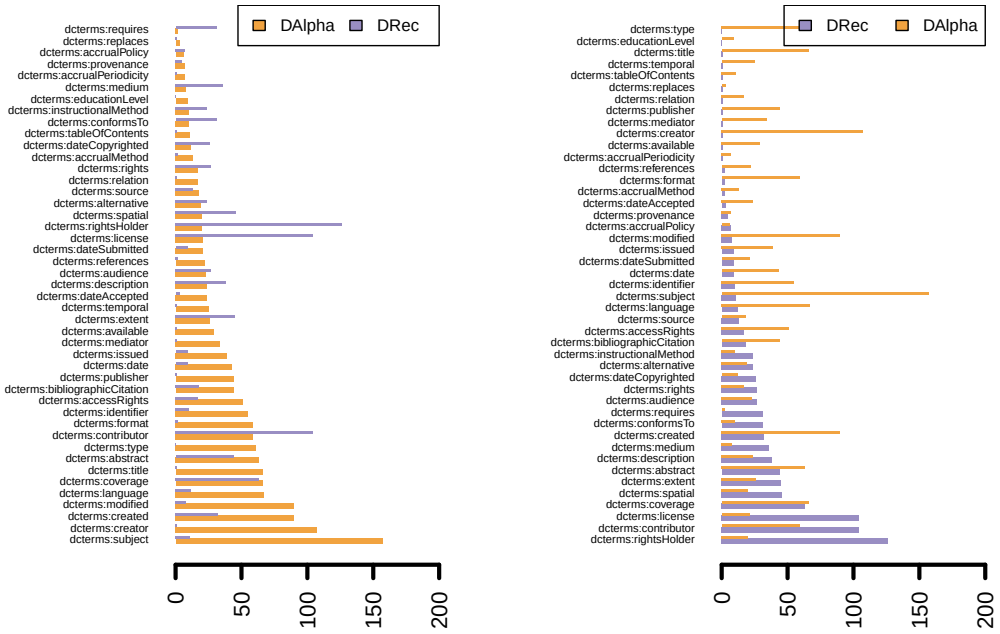

Figure 7: Distribution of descriptor instances, per descriptor, at the end of the user study

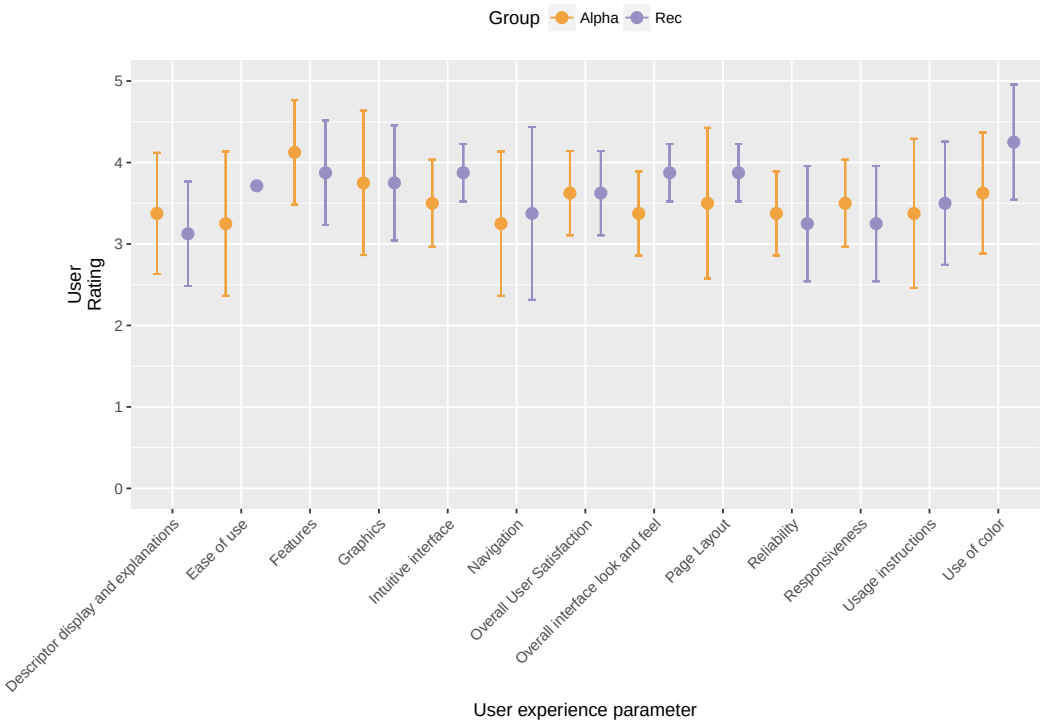

Figure 8: User ratings on the various surveyed parameters

while ranking the descriptors—perhaps ending up having a slight impact in response times and originating ocasional errors, disturbing some users. However, the average score still remained on the positive side and scores mostly over 3. In the "Ease of use" parameter, users found D_{Rec} better than D_{Alpha}, perhaps due to the partial automation of the metadata production (less scrolling, selection and browsing). Users of D_{Rec} found it easier to use overall than the users of D_{Alpha}. Given the number of additional features present in D_{Rec}, we expected that users would find it harder to use; since it was the opposite, there is anecdotal evidence here that users were able to learn the additional features and actually felt that they made their task easier. The user interface was considered reasonably intuitive by both user groups, with a slight advantage towards D_{Rec}, with a superior average as well as an inferior standard deviation. Finally, the user satisfaction of users was high, with all scores sitting between 3 (Satisfactory) and 4 (Good), the average being closer to 4.

Overall, we consider these results to be satisfactory for a software platform in prototype stage. Some missing functionalities were pointed out by users (moving folders and files, and file and folder renaming were the most frequent requests), but those missing capabilities did not prevent users from successfully carrying out their tasks in an efficient way. We are pleased to note that the average grades given by our users are 3 or above in all the analysed parameters.

7 Conclusions

With the strong push towards data repositories and data publication, namely by funding institutions, the practice of data description will have to evolve rapidly. RDM workflows and repository platforms need to accompany this change and adapt to their users. Therefore, even preliminary studies as we have presented can be valuable both to data managers at research institutions, who need to decide on the effort to invest in RDM, and to developers, who need to assess the effectiveness of the workflows promoted by their tools.

The more descriptors are made available to researchers, the richer and more diverse the descriptions can be. However, a large number of descriptors to choose from can overwhelm users without knowledge of data management. In the case of research groups that need to curate their own data, these issues can be even more prevalent. In this experiment with usage-driven descriptor ranking, Dendro is configured to use only Dublin Core descriptors, taking advantage of the fact that DC is arguably the most widely known and used schema in digital repositories.

The goal of offering the researchers a large choice of descriptors has to be balanced with convenience in the description task. The question is then: how can we assist the researchers (who are often the curators of their own data), to find the right descriptor to capture an important facet of their datasets? This is the rationale behind the idea of ranking descriptors according to their usage in the entire system, in shared collaboration areas (the *projects* in Dendro) and also by the specific user.

We have presented a descriptor ranking based on usage logs, where the interactions of users are classified according to categories that we have also specified. We then outlined the features that could help us rate the descriptors based on the usage interactions. Dendro has been extended to log user interactions and their types. With the available information, we designed formulas to calculate the components of an overall descriptor score. The parameters of these formulas were determined empirically to convey the relative importance of the different types of interactions that the users can perform in the system.

Our users were asked to participate in a study over 3 weeks, during which they had to use the platform to describe datasets from various sources. We were able to get a first picture of the influence that a descriptor ranking system has on a workflow supported by a research data management system. We have concluded that taking into account the past interactions of users with the data description system, we can reduce the number of interactions that users have to perform to fill in metadata records for research-related files and folders.

To evaluate the response of the users to the platform, we carried out a user satisfaction survey. The results indicate positive results (average rating of 3 and above) across all analyzed user interface parameters, which range from reliability to the overall look and feel of the platform and its user-friendliness.

The base data of these studies is available in a demonstration project at our Dendro demo instance[6] and Dendro is under active development as an open-source project on GitHub[7].

7.1 Lessons learned

This work was a good starting point for future experiments involving users and their interactions with a realistic data description system, and was also the first user test of Dendro. It was a first load test of the platform, since the users had to work simultaneously at least once a week. Some positive aspects of our experiment and some opportunities for improvement follow.

1. **Prepare your deployment**
   The first lesson learned is that after an experiment like this is set in motion, there can be no further modifications to the code running on the test instances, so the deployment has to be carefully executed. While

[6] See `http://dendro.fe.up.pt/demo`

[7] `https://github.com/feup-infolab/dendro`

bugs and possible improvements were identified by our users, we had to ask them to file them via email or in person during class meetings.

2. **Write a user guide but do not disclose too much**
   At the start of the experiment, users received a carefully written usage guide on the platform. At 3 pages, our short guide presented an overview of the features common to both Dendro instances, did not burden the students with too many details, and did not disclose the goals of the experiment. A point of contact was also provided in the guide so that they could report bugs or request assistance—bug report submissions and feature requests should be welcomed and encouraged.
3. **Make it worthwhile for your users too**
   The experiment was valuable not only to us as the researchers but also to the students. The quality of the finished descriptions was evaluated as part of the coursework, as well as a report about how to use Dublin Core to produce quality metadata. The results were very good, with the students achieving a median classification of `75%` with an average of `76%` and a `4,3%` standard deviation. Note that the person who conducted the experiment was not in charge of grading the students.
4. **Balance realism and the ability to study a variable**
   An aspect to improve in the future is the complexity of the interface presented to the users, which to a certain degree introduced additional variables in the experiment to consider a realistic system. An example is the button for switching between the "Automatic" and "Manual" metadata editor mode, which was not clear to the users (Figure 4). This probably caused manual selections to be more numerous than automated ones because the users simply did not notice the automatic descriptor features. We confirmed this later in some of replies given in the open answer section of the questionnaire sent to the users. To avoid situations like this, it is better to make modifications to the interface in order to isolate a single variable being studied instead of providing all the different options to the users.
5. **Think of alternative experiment scenarios**
   The design of the experiment can also be improved. While it is true that spliting the user sample *apriori* ensures the absence of learning bias, in our case the resulting subgroups became small. Instead, we could have had the pairs of students switch between Dendro instances throughout the work on each dataset source and have an even number of dataset sources. The initial adaptation stage might be carried out with a preliminary practice run on a dummy Dendro instance. This way, we could take advantage of the full user sample without having to split it from the start of the experiment.

7.2 Future work

This work is preliminary in many respects. To start with, ranking descriptors based on their usage requires the identification of relevant features, and this benefits from a somewhat long history of user interactions, which we have not available yet. Second, the selection of features is based on our intuition regarding their quality as predictors of descriptors relevance. Third, we used a convenience sample of users, namely a group of students with information science background and some basic familiarity with Dublin Core. The study highlighted a few design faults of the Dendro solution. Most were minor bugs that, while "somewhat annoying" in the words or our users, did not prevent them from carrying out their tasks.

A possible way to improve rankings is to include new features. For example, we want to make it possible for descriptors that were never used before by active user, or collaborators of the active project, to be on the list. We are also planning to include domain-specific metadata descriptors from ontologies other than DC. This approach will provide a novel solution for the usage-driven construction or improvement of Application Profiles, which is mainly a manual process carried out by research communities. While not completely replacing the curator's validation and oversight roles, these automatic data description capabilities can provide researchers greater autonomy in the description of their datasets, freeing curators from metadata creation to the tasks of validation and improvement of existing Application Profiles.

The logging system can have be made more complete to allow us to study more variables, for example the time until the selection or filling-in of the first descriptor after the webpage is loaded. This will allow us to examine the learning behavior of our users and compare the two instances.

A user-level quantitative analysis will provide more insight on why, for example, the U_{Rec} subgroup ended up saving a significantly lower number of descriptors than the U_{Alpha} counterpart. Our current conjecture is that one or more of the participants in the U_{Alpha} group have decided to perform a much more comprehensive description of their selected datasets.

Dendro has the ability to record the full history of metadata changes in any record. This makes it an adequate testing ground for studying the evolution of metadata records, so it can provide insight on how much effort curators need to put in to improve and maintain metadata records after their creation by researchers. We believe that such results may add to existing work that reports on the quality of submitted datasets containing DC-compliant metadata [16].

The "auto-complete" search box was used several times by the users to fetch specific descriptors. It may prove interesting to analyse the typing sequences of the values

input by the users to try to determine the terms that users search more often, and which part of the descriptor match leads to a successful selection (either the title or the comment property of the descriptor, as they both are matched when searching for descriptors).

The descriptor lists emerge from the actual usage of descriptors in the different research groups. In the long term, we believe that this information can be helpful for the formalization of an Application Profile (AP) [17], or to improve an existing one. The automated generation of the profile cannot replace the broad analysis and community-wide understanding that forms the basis of a Dublin Core Application Profile (DCAP) [28,29]. However, it may provide additional quantitative information regarding the changing needs of a community—which is an important driving force for the improvement of existing DCAPs [23].

In the context of the TAIL project, where the development of Dendro continues, we plan to broaden the scope of this work to additional metadata schemas besides Dublin Core. The usage-based approach can be applied to the description of datasets from diverse domains, helping users discover those sets of descriptors that best suit the specific nature of their datasets. While Application Profile is and should remain under the control of user communities, usage-based descriptor suggestions and rankings can provide additional data for communities to base their decisions on DCAP improvement.

8 Acknowledgements

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-016736.

References

1. J. Allinson, P. Johnston, and A. Powell. A Dublin Core Application Profile for Scholarly Works. *Ariadne*, 50:np, 2007.
2. R. Amorim, J. Castro, J. Rocha, and C. Ribeiro. A Comparative Study of Platforms for Research Data Management: Interoperability, Metadata Capabilities. *New Contributions in Information Systems and Technologies*, 1:101–111, 2015.
3. A. Ball. Scientific data application profile scoping study report. Technical report, UKOLN, University of Bath, Bath, UK, 2009.
4. S. Bechhofer, I. Buchan, D. D. Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, M. Gamble, D. Michaelides, S. Owen, D. Newman, S. Sufi, and C. Goble. Why linked data is not enough for scientists. *Future Generation Computer Systems*, 29(2):599–611, 2011.
5. T. Berners-Lee. Linked Data—Design Issues, 2006.
6. C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.
7. C. L. Borgman. The Conundrum of Sharing Research Data, 2011.
8. A. Boyko, J. Kunze, California Digital Library, J. Littman, L. Madden, Library of Congress, and B. Vargas. The BagIt File Packaging Format (V0.97), 2012.
9. K. Coyle and T. Baker. Guidelines for Dublin Core Application Profiles, 2009.
10. Dublin Core Metadata Initiative. DCMI Metadata Terms, 2012.
11. Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1, 2012.
12. European Commission. Guidelines on Data Management in Horizon 2020. (December):6, 2013.
13. V. V. D. Eynden, L. Corti, L. Bishop, and L. Horton. *Managing and Sharing Data: A guide to good practice*. UK Data Archive University of Essex Wivenhoe Park Colchester Essex CO4 3SQ, third edition, 2011.
14. A. Goy, D. Magro, G. Petrone, C. Picardi, and M. Segnan. Ontology-driven collaborative annotation in shared workspaces. *Future Generation Computer Systems*, 2015.
15. A. Green, S. Macdonald, and Robin Rice. Policy-making for Research Data in Repositories: A Guide. Technical Report May, DISC-UK DataShare Project, JISC, 2009.
16. J. Greenberg, S. Swauger, and E. M. Feinstein. Metadata Capital in a Data Repository. In *Proceedings of the International Conference on Dublin Core and Metadata Applications 2013*, pages 140–150, Lisbon, 2013. Data Dryad Repository.
17. R. Heery and M. Patel. Application profiles: mixing and matching metadata schemas. *Ariadne*, 25, 2000.
18. P. B. Heidorn. Shedding Light on the Dark Data in the Long Tail of Science. *Library Trends*, 57(2):280–299, 2008.
19. S. Hodson. ADMIRAL: A Data Management Infrastructure for Research Activities in the Life sciences. Technical report, University of Oxford, 2011.
20. R. Hu and P. Pu. Acceptance Issues of Personality-based Recommender Systems. In *Proceedings of the Third ACM Conference on Recommender Systems (Recsys '09)*, pages 221–224, New York, New York, USA, 2009. ACM.
21. L. Jahnke, A. Asher, and S. D. C. Keralis. *The Problem of Data*. Council on Library and Information Resources, 2012.
22. T. Joachims, L. Granka, and B. Pan. Accurately interpreting clickthrough data as implicit feedback. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154—161, 2005.
23. E. M. Krause, E. Clary, J. Greenberg, and A. Ogletree. Evolution of an Application Profile: Advancing Metadata Best Practices through the Dryad Data Repository. In *Procedings of the International Conference on Dublin Core and Metadata Applications 2015*, pages 63–75, 2015.

24. D. Lecarpentier, P. Wittenburg, W. Elbers, A. Michelini, R. Kanso, P. Coveney, and R. Baxter. EUDAT: A New Cross-Disciplinary Data Infrastructure for Science. *The International Journal of Digital Curation*, 8(1):279–287.
25. S. Leonelli, D. Spichtinger, and B. Prainsack. Sticks and carrots: encouraging open science at its source. *Geo: Geography and Environment*, 2(1):12–16, 2015.
26. P. Lord and A. Macdonald. Data curation for e-Science in the UK: an audit to establish requirements for future curation and provision. Technical report, JISC, 2003.
27. L. Lyon. Dealing with Data: Roles, Rights, Responsibilities and Relationships, 2007.
28. M. Malta and A. Baptista. State of the art on methodologies for the development of a metadata application profile. In *Communications in Computer and Information Science*. Springer-Verlag, 2012.
29. M. Malta and A. Baptista. A panoramic view on metadata application profiles of the last decade. *International Journal of Metadata, Semantics and Ontologies*, 9(1):58–73, 2014.
30. M. C. Malta and A. A. Baptista. A Method for the Development of Dublin Core Application Profiles (Me4DCAP VO.2): Detailed Description. *Proc. of the International Conference on Dublin Core and Metadata Applications*, (2009):90–103, 2013.
31. L. Martinez-Uribe. Using the Data Audit Framework: an Oxford case study. Technical report, Oxford Digital Repositories Steering Group, JISC, 2009.
32. L. Martinez-Uribe and S. Macdonald. User engagement in research data curation. In *Proceedings of the 13th European conference on Research and advanced technology for digital libraries*, volume 5714, pages 309–314. Springer, 2009.
33. H. Piwowar and T. Vision. Data reuse and the open data citation advantage. *PeerJ*, 1:e175, 2013.
34. H. A. Piwowar, R. B. Day, and D. S. Fridsma. Sharing detailed research data is associated with increased citation rate. *PLoS ONE*, 2(3), 2007.
35. J. Rocha, J. Castro, C. Ribeiro, and J. Correia Lopes. Dendro: collaborative research data management built on linked open data. *Proceedings of the 11th European Semantic Web Conference*, 2014.
36. J. Rocha, J. Castro, C. Ribeiro, and J. Correia Lopes. The Dendro research data management platform: Applying ontologies to long-term preservation in a collaborative environment. In *iPres 2014 Conference Proceedings*, 2014.
37. J. Rocha, C. Ribeiro, and J. Correia Lopes. Managing research data at U.Porto: requirements, technologies and services. *Innovations in XML Applications and Metadata Management: Advancing Technologies*, IGI Global, 2012.
38. J. Rocha, C. Ribeiro, and J. Correia Lopes. Ontology-based multi-domain metadata for research data management using triple stores. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, 2014.
39. A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02)*, 46:253–260, 2002.
40. Science Magazine. Dealing with data. Challenges and opportunities. Introduction. *Science (New York, N.Y.)*, 331(6018):692–3, feb 2011.
41. R. Sinha and K. Swearingen. Comparing Recommendations Made by Online Systems and Friends. *DELOS Workshop on Personalisation and Recommender Systems in Digital Libraries*, 2001.
42. R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *Extended abstracts on Human factors in computing systems (CHI '02)*, page 830, 2002.
43. S. Strickroth and N. Pinkwart. High quality recommendations for small communities: the case of a regional parent network. *Proceedings of the sixth ACM conference on Recommender Systems*, pages 107–114, 2012.
44. K. Swearingen and R. Sinha. Beyond Algorithms: An HCI Perspective on Recommender Systems. *ACM SIGIR 2001 Workshop on Recommender Systems (2001)*, pages 1–11, 2001.
45. M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3, 2016.