International Journal on Digital Libraries
https://doi.org/10.1007/s00799-018-0238-x

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]

Received: 31 January 2017 / Revised: 29 March 2018 / Accepted: 13 April 2018

Abstract
Dublin Core descriptors capture metadata in most repositories, and this includes recent repositories dedicated to datasets. DC descriptors are generic and are being adapted to the requirements 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 best 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, (3) users were satisfied with the performance of the platform, and (4) the quality of description was not hindered by descriptor ranking.

Keywords Research data management · Metadata · Dendro · Descriptor ranking · Dublin core · Usage information · Dc terms · Collaborative · Linked open data

1 Introduction

Data are becoming increasingly important research outputs, especially as research moves toward the 4th Paradigm of Science, where research is powered and supported by increasingly large amounts of data and computational power [21]. Moreover, the movements toward open data and open science are creating strong incentives for researchers to organize and publish their data [29]. Data curation, providing safely stored, properly described and uniquely identified datasets, emerges as a competence in libraries and research centers, as the professional profile for data curators becomes delineated.

✉ João Rocha da Silva
joaorosilva@gmail.com

Cristina Ribeiro
mcr@fe.up.pt

João Correia Lopes
jlopes@fe.up.pt

[1] INESC TEC, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

Making data available under an open license is expected as the default in research, with the necessary provision for regulated and sensitive data. Research data management now refers to actions that range from the support to data creation in e-science environments to long-term preservation in archival repositories, and including persistent identification, description and reuse.

Researchers need to share data, so informal data sharing takes place in spite of the lack of appropriate support for data curation [45]. There is a demand for datasets, both as basic outputs and as a complement to research papers. On the other hand, high-quality metadata are vital for the discovery, retrieval and interpretation of research data [31]. Metadata are also crucial to enforce the so-called FAIR principles, proposed as a framework for data access and reuse [53]. As data are recognized as research outputs, data citation becomes a central issue. Researchers build upon their peers' results for the advancement of science, and the proper credit to data is essential [38,46].

While informal metadata are commonly created as researchers gather their datasets [25,43] and used to share data

Published online: 26 April 2018

within research groups [36], the production of metadata that are good enough for sharing with the community requires much more effort and a higher degree of knowledge of metadata practices [8]. In an ideal scenario, researchers would work together with data curators to adequately capture the production context of a dataset [32]. In the current state of affairs, however, curators are unavailable in many research groups and there are few financial resources for data curation. This is especially true on the so-called long tail of science, the large numbers of small research groups that produce a total amount of data comparable to that of the small number of large research groups [20]. As a result, many researchers have to describe and publish their datasets with limited curatorial support.

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 favor exchange and aggregation, and ease of use, so that researchers can focus on the description 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 [40]. 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 as ontologies. However, the more descriptors are available, the harder it is to select the right descriptors for a user, a research group, or a domain. This is the rationale behind the idea of ranking descriptors according to their past usage. In the proposed approach, Dendro uses information such as past user behavior, the currently open project or the currently logged in user to help its users discover and use adequate descriptors for a dataset.

To test the ideas on usage-based descriptor ranking, we have set up an environment where the interactions of users with Dendro are recorded. To limit the scope of descriptors for this experiment, Dendro is configured to use only Dublin Core Metadata Terms (DCTERMS) descriptors [11]. Dublin Core is the most widely used vocabulary in digital repositories, with well-established meaning for the descriptors, 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 or their research groups.

The paper is organized 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 presents the descriptor ranking approach, namely the selected features and the observations used to quantify them. The details of the user study are presented in Section 5. Section 6 analyses the data collected in the study and is followed by the conclusions.

2 Research data management workflows

The context of this work is the promotion of data management and data publication in small research groups with no specialized data curation support. We are working with several research groups from the University of Porto with the goal of setting up a complete workflow for managing research data from the start of research projects, in compliance with national and international data mandates [13].

There are many questions that arise during our meetings with 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. By metadata model, we mean a set of descriptors, possibly coming from different vocabularies, which is available in a machine-processable representation. A metadata model 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 promote systems interoperability, allowing for the aggregation of datasets and improving discoverability.

We want to build metadata models that suit the description needs of specific research groups, including generic descriptors such as Dublin Core and domain-specific ones. The definition of clear research data management workflows makes 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 researchers invest in preparing a dataset for publication, it is important to make its metadata available as widely as possible, and to keep them associated with the data 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 [53], but also with the best practices in digital preservation [24].

2.1 Complexity vs. capability in DC metadata

The Dublin Core Metadata Element Set (DCMES) [12] is still widely used due to its simplicity, maturity and interoperability, even though some of its shortcomings were identified a long time ago [1]. The DC elements were originally intended for lay users, and concepts such as the allowed

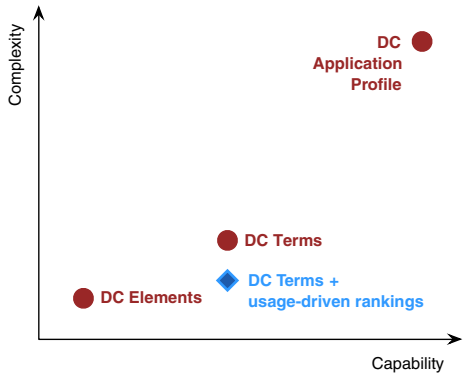

Fig. 1 Tradeoffs in Dublin Core metadata

range for a descriptor were not included. One of the results was ambiguity regarding the appropriate 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 [11]. The representation of DCTERMS in Resource Description Framework (RDF) enables the publication of metadata records as Linked Open Data [7], highlighting the importance of ontologies for the exchange of metadata records between systems [5]. 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.

Dublin Core Application Profiles (DCAP), a concept derived from that of an Application Profile [19], are the next step toward 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 consider an adaptation period [1,10,34]. After the DCAP is in production it can also evolve according to the changes in data models of prominent repositories [27].

Figure 1 shows a coarse visual comparison of the trade-off between the complexity of the three alternatives and their ability to comprehensively represent metadata. DCMES is very easy to understand but has obvious limitations in expressive power. DCTERMS provides a much more detailed metadata model using 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 data. 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 research teams often change as projects come closer to completion, making it unlikely for researchers to engage in the production of metadata for data they are no longer working with. 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 [4,14,37]. This need has been identified in the past by projects such as ADMIRAL [22], 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) [28].

Dendro[2] brings dataset deposit and description to earlier stages 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 [42]. Users start by creating a Dendro project, which is a storage area shared among project contributors. Inside each project they can upload files and create folders, while collaboratively producing metadata records for each of them [40]. One important feature of Dendro is that it takes advantage of existing data repositories to make described datasets available in the long term. Dendro interoperates with these platforms, offering dataset export plugins for CKAN, EUDAT B2SHARE, Zenodo and figshare, while support for others can be provided through additional plugins.

Dendro has an extensible data model built on ontologies, allowing users to easily "mix-and-match" descriptors from different schemas in their metadata records. We have compared this model to those present in the relational database

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

[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.

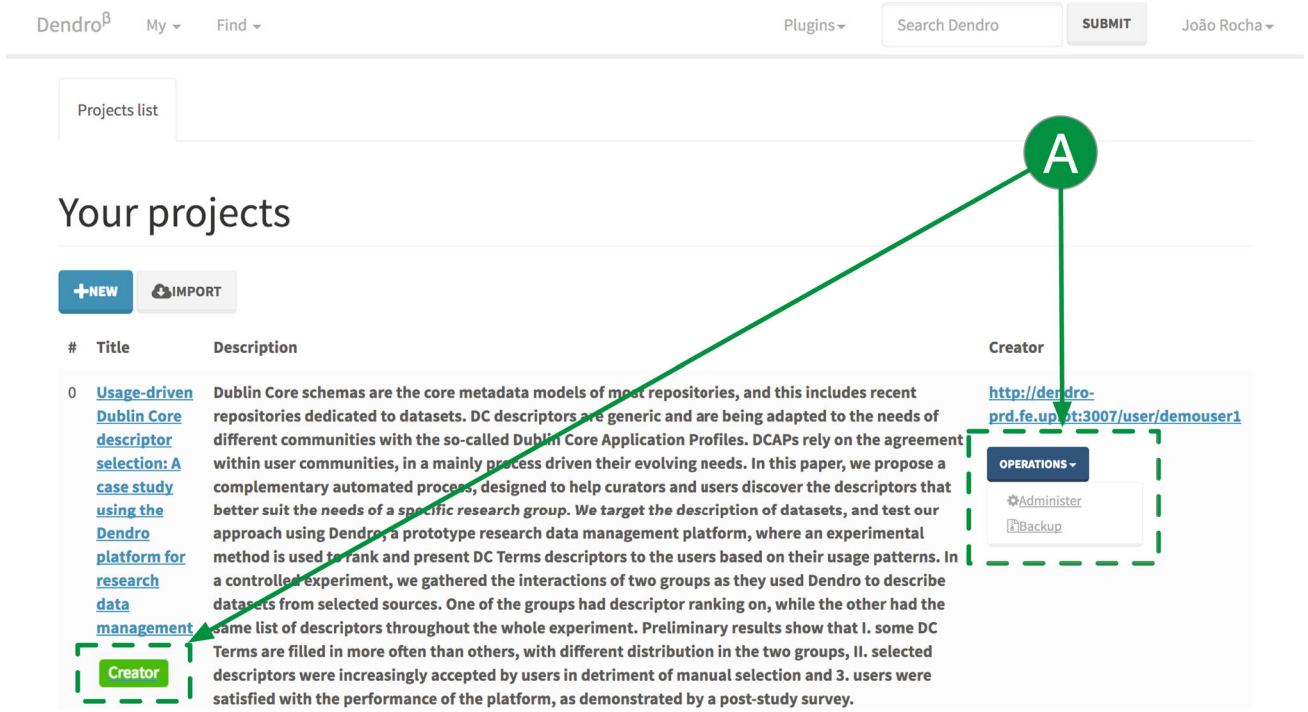

Fig. 2 The list of projects in Dendro

schemas that support the most widely used open-source repository platforms, and concluded that a graph-based data model allows the efficient representation of directory structures (relationships between parent and child nodes), where the nodes have attributes (metadata descriptors) that are unknown at the time of database modeling and whose values need to be versioned. The other platforms, based on relational models, compromise by not allowing datasets to have a file hierarchy, having fixed metadata sets, or not having metadata versioning. Dendro satisfies all requirements simultaneously because of its flexible graph data model [41]. On the subject of interoperability, other repository platforms often implement protocols such as OAI-PMH or specific REST APIs to expose their metadata records to external systems [3]. Since it uses a graph database, Dendro transparently provides URI dereferentiation, as proposed by the Linked Data guidelines [6] and a SPARQL endpoint, which allows for much more sophisticated querying than the OAI-PMH protocol.

3.1 A brief overview of Dendro

Dendro allows users to create projects, which are very similar to the "shared folders" of Dropbox. Users can see the projects where they 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 for listing the projects that the current user has access to. The project creator will have access to 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 [9] 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 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 a metadata read-only layout, optimized for viewing.

All project members can describe files or folders in the project's directory structure using the metadata editor, as shown in Fig. 4. The layout is divided into three main sections, from left to right: the file browser, where users browse their files and folders; the metadata editor, showing the current descriptors and controls for users to edit their 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 another to "Hide" that descriptor. When the user selects the "Favorite" button for the first time, the descriptor will be marked as "Project Favorite"

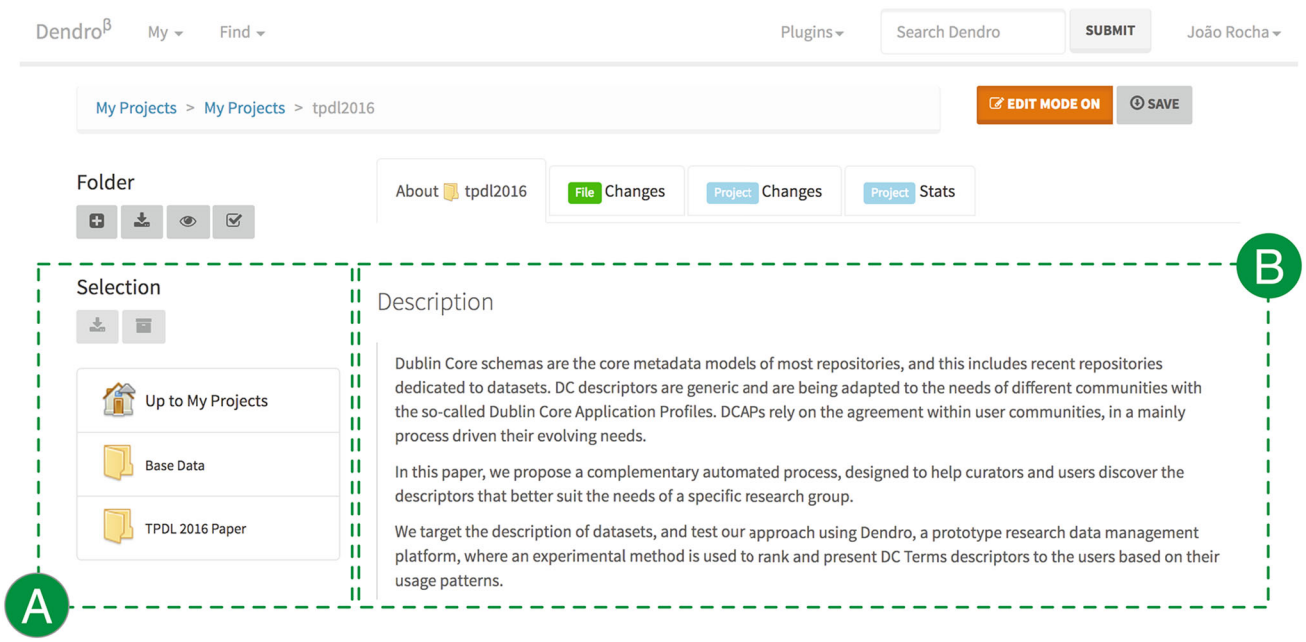

Fig. 3 Viewing the root folder of a Dendro project

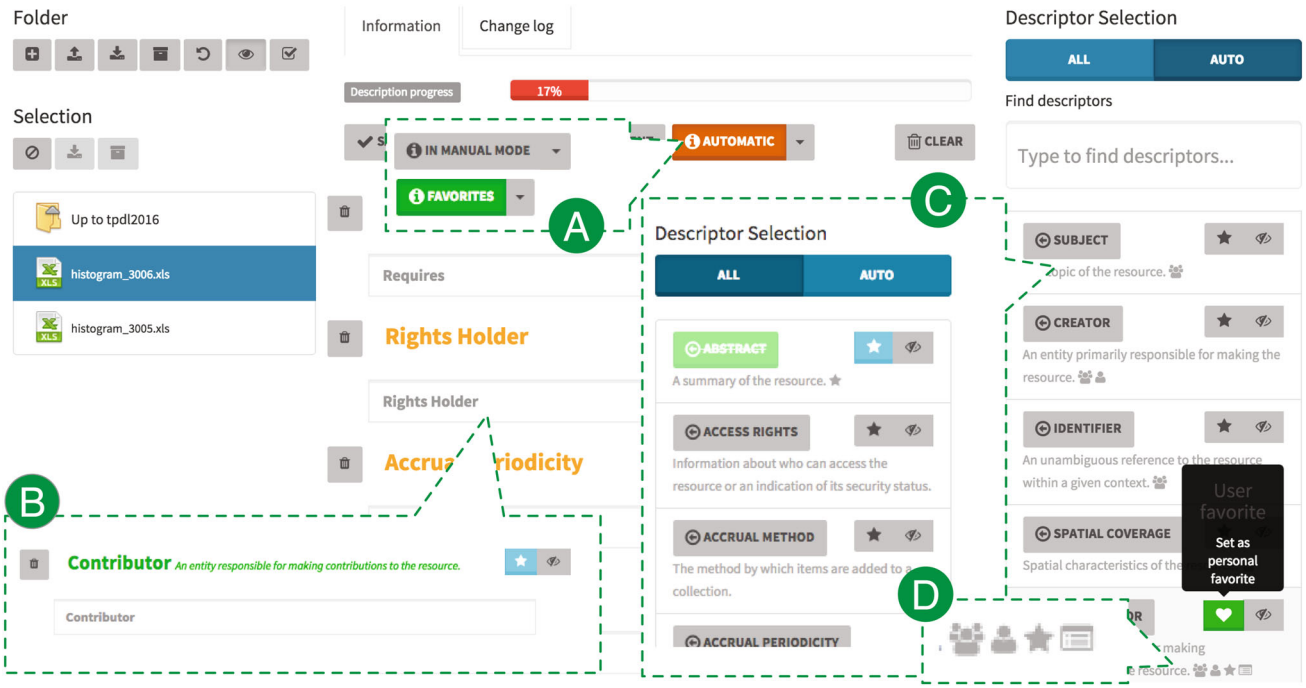

Fig. 4 The main user interface of Dendro

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.

Area **A** shows the interface modes available to the users. 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 center, 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 (**D**), in order to improve system *transparency* [48,51]. For each descriptor in area **C**, a set of icons may 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").

4 Descriptor ranking approach

The main motivation behind the ranking of descriptors is to bring to the attention of the researchers the descriptors they are likely to adopt. The data model of Dendro makes it possible to combine generic and domain-specific descriptors. However, for this experiment we test our descriptor ranking approach, using only descriptors from the DCTERMS[3] ontology.

The descriptor ranking approach explored here has been proposed originally in a context where several ontologies are available to the users, and therefore managing the complete set of descriptors is infeasible in a data description session [39]. This section provides an abridged form of the justification for the set of features used in ranking, as well as their weights.

4.1 User interaction logs

To capture descriptor usage, Dendro was extended with several interaction logging capabilities. These logs provide evidence of descriptor usage in several modalities and 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. Every record represents a user interaction, which is a piece of evidence of descriptor use or intention of use. The first column (`uri`) is the identifier of the interaction record. In the graph model of Dendro [41], every resource (including interaction records) has a unique identifier. In this case, the identifier has the form of an URL, thus allowing the dereferentiation 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 unique identifier of the user who performed the interaction. The type of each interaction is represented by the `type` column, and is the kind of action that the user performed. The types of interactions have been identified according to the different actions that users may take in the course of a description session. The `rankingposition` of the interaction is relevant for some types of interactions and records the position of the descriptor in a list at the time of that interaction. For example, when a descriptor is selected from the list on area **C** of Fig. 4, we record its position in the list. Conversely, a value of -1 means that the position is meaningless for the type of interaction in that entry. For example, 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 toward a single descriptor. Users tend to type more characters until they 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 `rdf:label` or `rdf:comment` properties, allowing users to retrieve descriptors even if they do not know their exact name. The types of interactions monitored by Dendro are shown in Table 2; every interaction type has a name and a brief description.

4.2 Descriptor features

A descriptor *feature* is a characteristic of the 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 ability to assess the degree of preference toward a descriptor. Features capture aspects such as which descriptors are filled in or selected, by which user, at what time, under which project, and the type of resource being described.

The first features in Table 3, f_1 to f_4, capture implicit feedback from the user. For example, filling in a descriptor is implicit feedback because it expresses an 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 will contribute differently to the score of a descriptor, as they express different degrees of preference toward 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 score

[3] http://bloody-byte.net/rdf/dc_owl2dl/dcterms.

Table 1 Excerpt of the interactions log

url	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

used in descriptor ranking. The overall score of a descriptor is the sum of these components, calculated from user interactions. The formulas proposed here for the components and for the descriptor scores are intentionally very simple, to make their effect on the lists of available descriptors easy to interpret.

To calculate a score value for a descriptor d, we define the n *components* of its score $c_{i,d}$ with $i = 1 \ldots n$. For each feature, the component is the result of a formula that takes as input a measure associated with the feature. Feature f_1, the number of times a descriptor is used, contributes to the score of descriptor d as $c_{1,d}$.

The score S_d for descriptor d, is computed using equation 1. It is simply a sum, over all features, of the values of the n components.

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

Table 4 details the components and the formulas adopted for each.

The formulas for the components were set empirically, and 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. Also, a single interaction may contribute to more than one feature—for example, accepting a descriptor within a project contributes to f_1 (frequent use of the descriptor overall) but also to f_3 (frequent use 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 these accumulated frequencies by 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 means to avoid certain features from overwhelming others due to their weights. For feature f_1, for example, the component is

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

The components for features f_2 to f_4 are similar. The component for f_5 is calculated via textual similarity[4] obtained from the search engine used in Dendro, Elasticsearch [15]. 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,

[4] https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-mlt-query.html.

Table 2 Types of interactions monitored in this experiment

Interaction type	Description
`accept_descriptor_ from_autocomplete`	Selecting a descriptor from the autocomplete box in **B**
`accept_descriptor_ from_manual_list`	Selecting 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`	Selecting 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`	Filling in a descriptor that was 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 is filled after they are added, an interaction of this type is 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 row, but the user clicks the "Previous Page" button at the top of the descriptor list instead of the "Next Page" one
`favorite_descriptor_ from_quick_list_for_project`	Marking 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 row, 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 (**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 row, but for user favorites

requires the user to stop the description tasks to explicitly change descriptor status, so they should 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 users mark them as hidden for themselves or for the current project (features f_{10} and f_{11}).

The main reason for 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 [44]. Other motivations are the limited timespan of the experiments and the size of our user sample, which would 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 in 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 systems that require information management expertise—in fact, some are already information science professionals.

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 Fig. 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?

This is considered a small-scale study, justified by the lack of similar studies in this area, the fact that the use of data repositories still has to be established as a regular research activity and that, even as this use increases, the difficulty in assembling usage data also becomes greater due to the diversity of domains. Even in more well-established areas such as Information Retrieval and recommendation, small-scale user studies are justifiable [16,26,49].

A small-scale study has obvious limitations and does not provide evidence for general results. A trade-off we faced in this case was realism versus control over experimental variables. We wanted to use the real Dendro interface that has many features not related to the experiment we designed. This, however, creates dependencies between variables and makes results more difficult to interpret. We will elaborate on that in the analyses and the conclusions.

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. This allowed 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} (using the Dendro with descriptors ordered alphabetically). Students were randomly assigned to U_{Rec} or U_{Alpha}. Each student was tasked, over three weeks, with the description of several datasets collected from different online sources and belonging to distinct research areas. Three sources of datasets were used: ICPSR[5], B2SHARE[6] and re3data[7]. In the case of re3data, which is a repository registry, 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 where people collaborate in projects and discuss the most convenient metadata to assign. Nevertheless, each participant had the same number of individual tasks.

While using Dendro, the interface elements in areas **A**, **B** and **D** (see Fig. 4) were only available for participants in

[5] https://www.icpsr.umich.edu/icpsrweb/.

[6] https://b2share.eudat.eu/.

[7] http://www.re3data.org/.

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 Fig. 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)

U_{Rec}. The **C** elements were available to all, but the "Auto" button was not enabled for users in U_{Alpha}.

6 Results analysis

To evaluate the effects of descriptor ranking in the description process, we carried out several complementary analyses. First, we set out to determine if the descriptions produced by Dendro users are adequate for the corresponding resources. Afterward, we evaluate whether the introduction of descriptor ranking can save description effort and reduce user inaccuracies, while maintaining the quality of the finished descriptions.

6.1 Metadata record quality

To measure the quality of the metadata records produced by the users, we gathered a "ground-truth" dataset (GT) against which to compare those records. We already mentioned that our experiment went along three stages, with datasets from ICPSR, B2SHARE and re3data being described in successive stages. We decided to consider only one of the sources, the ICPSR, for our GT. While including records from the three sources in the GT would add realism, it would also introduce additional variables in our study, since metadata quality and comprehensiveness can vary greatly across repositories and datasets. This repository was selected not only because it is considered a reference data source for research in social sciences, but also because it holds a Data Seal of Approval (DSA) certification since May 2014, under the DSA 2014-2017 guidelines[8]. The DSA specifies a comprehensive set of requirements for all steps of the data management workflow, including metadata production. ICPSR datasets are manu-

[8] https://assessment.datasealofapproval.org/assessment_114/seal/html/.

ally curated and thoroughly reviewed for completeness after deposit [50] and ICPSR data managers contact data producers to obtain the required information if it is found to be missing [52].

In the sequel, we will use the notation R_{Dendro} to designate a metadata record as completed by one of the participants, i.e., a set of <descriptor, value> pairs for the descriptors selected by the participant. Similarly, R_{GT} will designate a similar record as obtained from ICPSR. The participants using the *Alpha* version produced a set of records denoted as D_{Alpha} and those using the *Rec* version the set D_{Rec}.

To prepare the ground truth, we looked at the Dendro descriptions collected in the ICPSR part of the experiment and manually matched each with the corresponding ICPSR dataset URI, using either the record DOI or other metadata such as title or abstract. From this analysis, we identified records for 14 distinct ICPSR datasets in D_{Alpha} and 37 for D_{Rec}. Their corresponding DC records were then harvested programmatically via the ICPSR API to build the GT dataset.

Each Dendro record R_{Dendro} produced in the description stage of the experiment and its GT counterpart R_{GT} will use some of the available descriptors. Let us consider one of these descriptors d_j. d_j will have descriptor instances $d_{j,Dendro} \in R_{Dendro}$ and $d_{j,GT} \in R_{GT}$.

To determine how close $d_{j,Dendro}$ is to $d_{j,GT}$, we calculated the Jaccard similarity $J(d_{j,Dendro}, d_{j,GT})$ between the sets of words in $d_{j,Dendro}$ and $d_{j,GT}$. Then, if $J(d_{j,Dendro}, d_{j,GT}) > 0.5$, we considered d_j to be correctly filled in. The Jaccard similarity was selected because of its simplicity and ease of understanding, and also because it is not influenced by the order of the words in the texts being compared. This is important to account for cases where there was more than one instance of a descriptor d_j in the same record (e.g., more than one `dc:subject` for the same record R_{Dendro}). In these cases, this choice of similarity measure allowed us to concatenate all values into a single descriptor instance before tokenizing the descriptor value into its words[9].

Figure 5 shows the percentage of descriptor instances that match their GT values for D_{Alpha} and for D_{Rec}. A plain application of the distance metric results in an overall percentage of correct descriptors near 50% in both cases (see the last row in Fig. 5). This result, although positive in this context, called for further enquiry regarding the descriptors where no similarity was present. These were important descriptors, such as `dc:type`, `dc:date`, and above all `dc:description`. To further analyze descriptor usage, we have performed another similarity analysis, this time calculating, for each Dendro instance, the Jaccard similarity for each pair of descriptor instances $d_{j,Dendro}$ and $d_{k,GT}$, one from the user descriptions, the other from the corresponding GT record, regardless of the descriptor.

As shown in Figure 6, there is a very high average similarity between the values of the `dc:abstract` descriptor present in Dendro records and the `dc:description` present in their GT record—this is a clear indication that participants systematically used the `dc:abstract` descriptor to represent the `dc:description` of the dataset. There are other similar decisions, such as, in D_{Alpha}, the use of `dc:rightsHolder` for what is `dc:creator` in the GT, and `dc:type` mistaken for `dc:source`. In the case of D_{Rec}, we find `dc:issued` used for what is `dc:date` in the GT. Another conclusion that can be drawn is that D_{Alpha} users adopted a wider range of descriptors in their records. This is illustrated by the presence of more possible values (20 vs. 18) in the y-axis in Fig. 6.

In light of this analysis, and returning to Fig. 5, we can conclude that the quality of the descriptions produced in D_{Alpha} and in D_{Rec} is quite good. If we ignore the different choices of descriptors in our analysis, e.g., by leaving out `dc:description`, the overall quality of the descriptions greatly increases. Another source of disagreement with the GT comes from descriptors with values strongly influenced by representation formats: a `dc:date`, for example, can represent the same value in a Dendro record and in its GT counterpart, but the representation in different locales leads to completely different literal values and thus to low similarity. The `dc:identifier` descriptor suffers from the same problem, with the use of a mix of URI, DOI and ICPSR internal IDs. While they all match the DC definition of identifier ("An unambiguous reference to the resource within a given context"), format differences can hide their similarity. The 3 lines at the bottom of Fig. 5 show averages for correct descriptors disregarding descriptors where different choices were considered in R_{Dendro} and R_{GT}. More than 60% agreement is present if we ignore `dc:description` and nearly 75% if `dc:date` and `dc:identifier` are ignored instead. These percentages would obviously be even higher if these descriptors were considered as correctly represented, rather than ignored.

6.2 Descriptor selection accuracy

To determine if the introduction of descriptor ranking helps the users select the appropriate descriptors (as per the GT), we analyzed the presence of GT descriptors in every record of D_{Rec} and D_{Alpha}. To do that, we computed the Jaccard similarity between the sets of descriptors included in each record by the users and the sets of descriptors used in the corresponding GT. For example, if for a record in R_{Dendro} a user selects and fills in `dc:title` and `dc:abstract` (regardless of their actual values) and the corresponding R_{GT}

[9] We have used the `tokenize_words` method of the `tokenizers` R package, available at https://cran.r-project.org/web/packages/tokenizers/index.html.

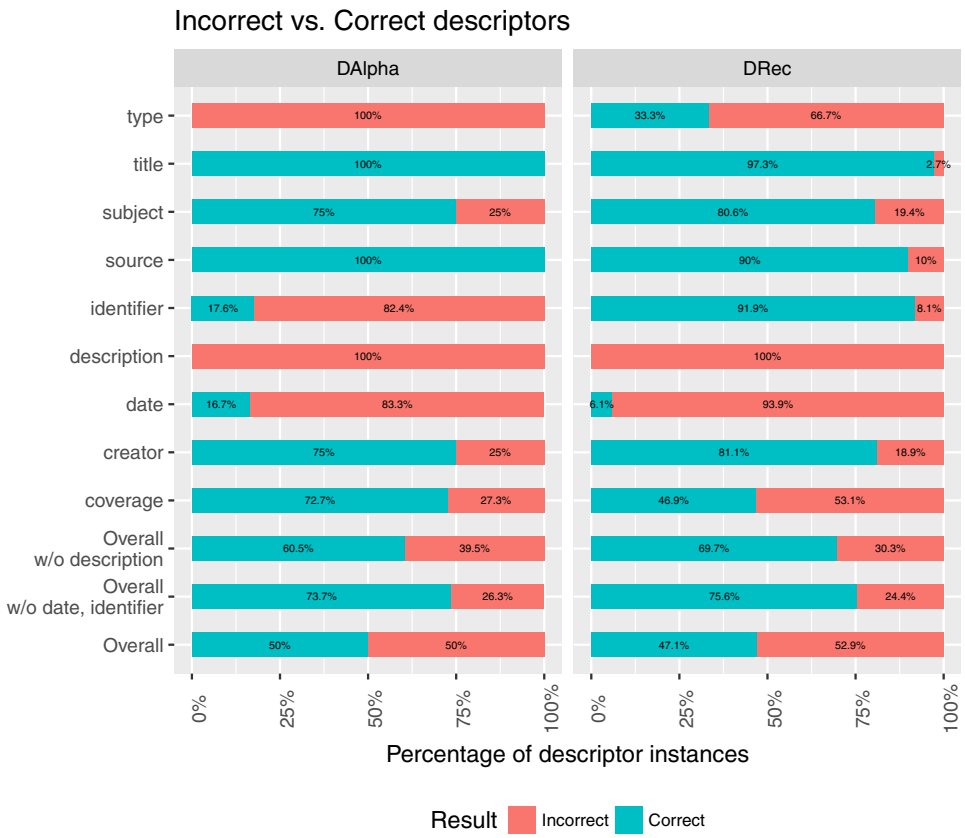

Fig. 5 Correct versus incorrect descriptors

record includes `dc:title` and `dc:description`, their similarity $J(R_{Dendro}, R_{GT})$ would be $1/3$.

After calculating all the similarities for the 14 distinct ICPSR datasets in D_{Alpha} and 37 for D_{Rec}, we calculated their averages, $\overline{J}_{D_{Alpha}} = 0.28$ and $\overline{J}_{D_{Rec}} = 0.45$. In order to determine if the results are statistically significant, we used a t test for two independent samples, concluding that the average similarity for D_{Rec} is higher than that for D_{Alpha} ($t = 5.7809$, $df = 22.013$, $p - value = 8.124e - 06$).

6.3 Description effort

The analysis of the logs resulting from the user study has to consider the interactions of the users in several tasks. For the remaining analyses, we consider the whole set of descriptions, and not only those originated in an ICPSR dataset.

We define *effort interactions* as those interactions that users 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 Fig. 4). Conversely, those interactions that do not require any effort toward the descriptor lists are *non-effort interactions*. For example, Fig. 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 filled in and saved.

Figure 7 shows a comparison of the average position of the descriptors as they were selected from the list. Lower values indicate an average position closer to the top of the list; the lower the value the better the ranking performs, since users find what they need at 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 Fig. 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

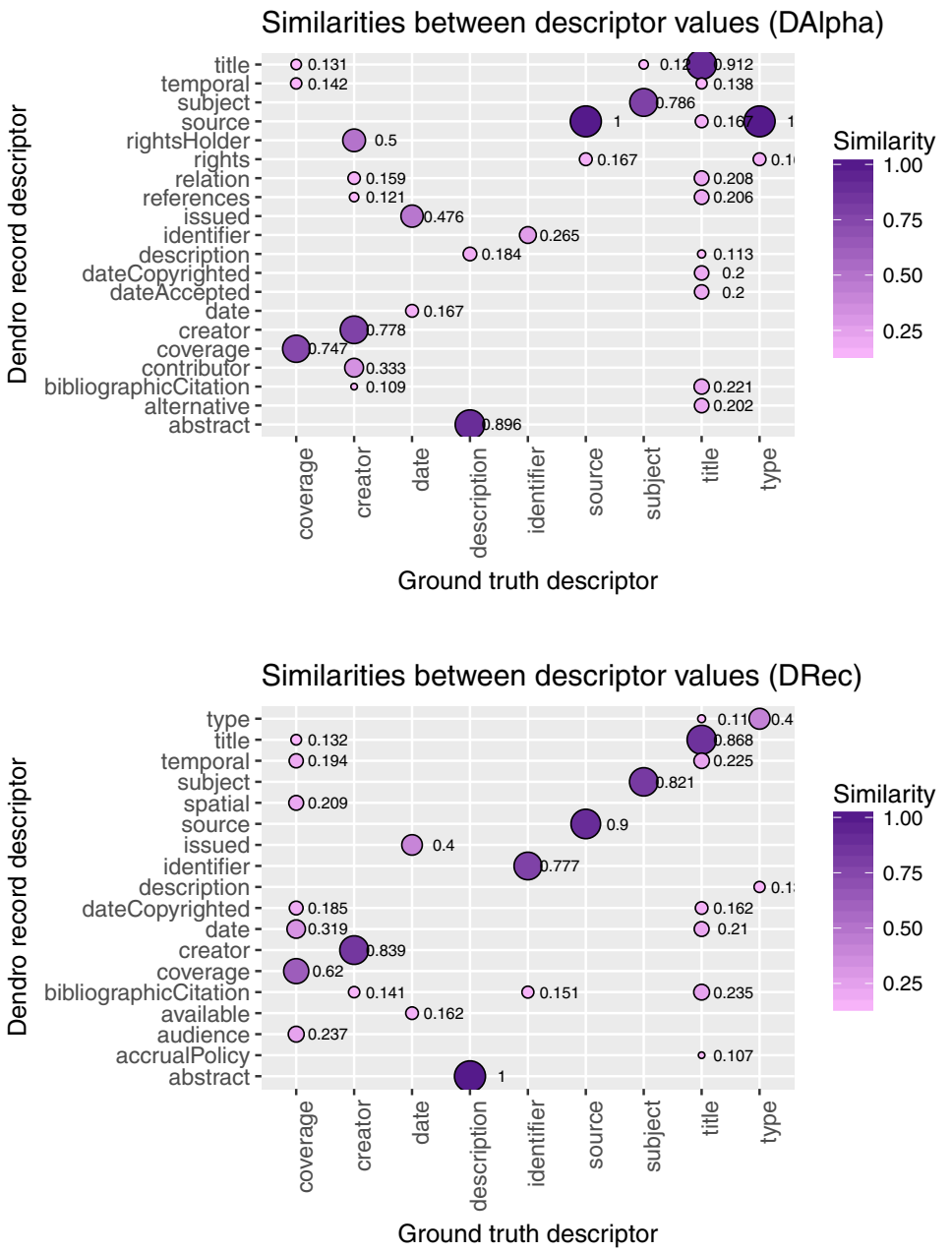

Fig. 6 Descriptor–descriptor similarities

when the "AUTO" option is selected in D_{Rec} (see Fig. 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 also analyzed the number of effort interactions

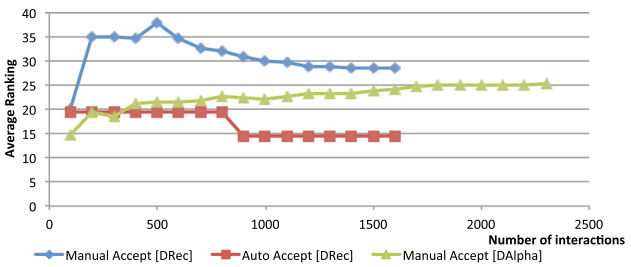

Fig. 7 Average position of the selected descriptors in the selection lists

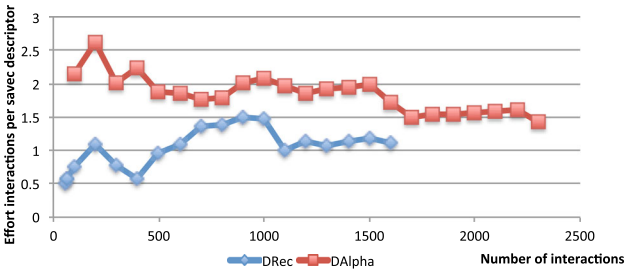

Fig. 8 Effort interactions per saved descriptor

for every descriptor that is filled in. Figure 8 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 interactions 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 9 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, the top-n descriptors in D_{Rec} concentrate a number of descriptor instances larger than the top-n 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 tend to 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.

6.4 Number of descriptor edits and deletions

Since Dendro records all revisions and the types of changes made to the record for every descriptor changed (added, edited, removed), it is possible to analyze the revision history of all records. Whenever a descriptor is added to a metadata record, a new "change" with type "Add" is recorded. If the descriptor is edited at a later time, another change of type "Edit" is recorded. If the user deletes the descriptor entirely, a change of type "Delete" is recorded in Dendro. In a perfect scenario, users would describe their datasets with the best descriptors and the right values for those descriptors, without the need for any corrections. That would mean 100% of the descriptor changes would be of type "Add".

Analyses of descriptors that require the most corrections have been performed in the past to help data curators and institutional managers pinpoint flaws in data deposit workflows [17]. In this case, we are interested in reducing the additional effort incurred by such corrections.

Figure 10 shows, for each descriptor d, the percentage of additions, edits and deletions. The first important aspect to point out is that some bars are missing in the $DRec$ area; this is due to the fact that those descriptors were not used at all in that instance. A high number of corrections may indicate that the users are very concerned about the quality of their metadata records, to the point of reviewing them multiple times. While editing is a sign of improvement, deleting a descriptor can be an indication that the problem was not the value of the descriptor but the descriptor type that was not correctly chosen. From the chart it is apparent that there is a slight reduction in the ratios of "Edit" (18% in $DAlpha$ to 11% in $DRec$) and "Delete" changes (9,4% in $DAlpha$ to 4,4% in $DRec$). This means that, on average, the users of $DRec$ filled in their metadata records with slightly less correction effort.

6.5 User satisfaction survey

To complement our quantitative analysis, we performed a user satisfaction survey in line with similar work [49]. Our user survey was anonymous; we got answers from 18 users out of the initial 23 participants, 9 from each Dendro instance. From the 18 respondents 12 were women and 6 men, with a median age of 23 years, average age of

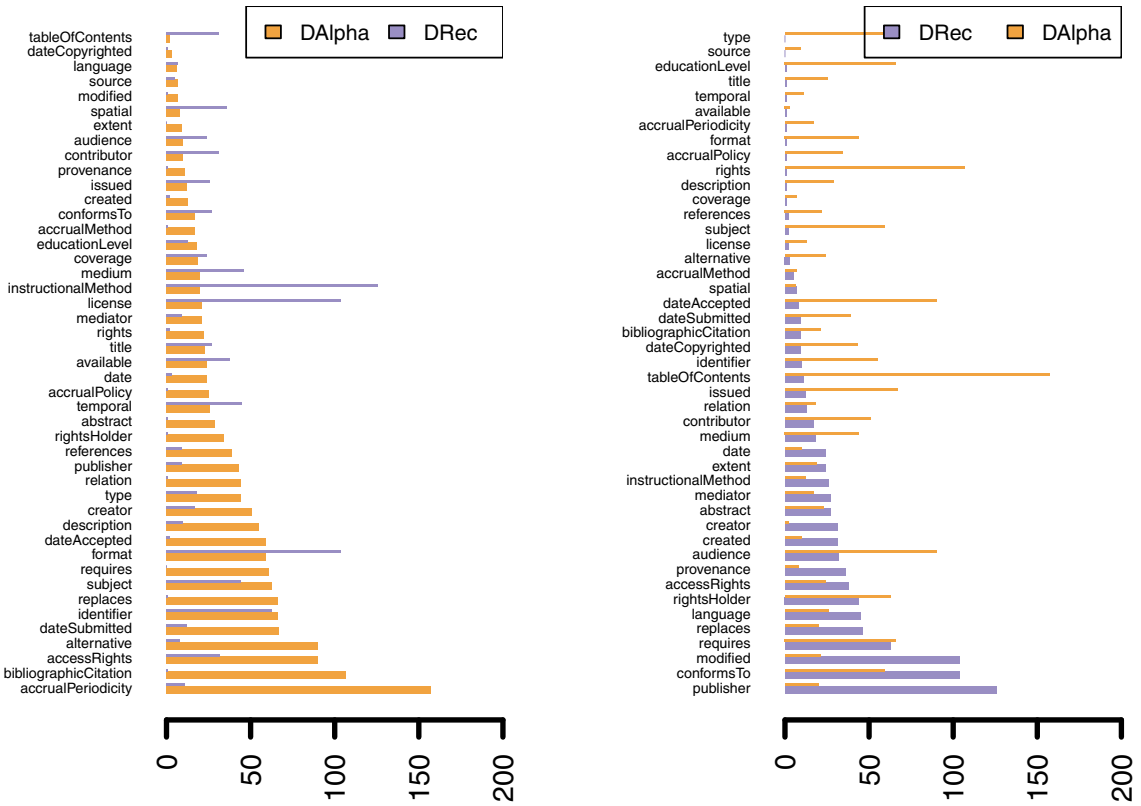

Fig. 9 Distribution of descriptor instances, per descriptor, at the end of the user study

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 [23,47], we outlined a series of questions specific to our case, which covered interface aspects—such as "Page Layout" or "Use of color". We reused some categories proposed in these studies, but adopted a 1–5 star scale through which users rated each aspect.

The results of the survey are shown in Fig. 11; 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 color, as most answers reside in the $[3 - 4]$ interval and the average is close to 4 versus an average score of 3.5. As shown in Fig. 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 colors indicates that this information is useful to users and does not overload the interface.

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 a slight advantage toward the D_{Rec} instance, perhaps due to the additional functionalities that were present, which were always accompanied by elements aiding the user, 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 "Respon-

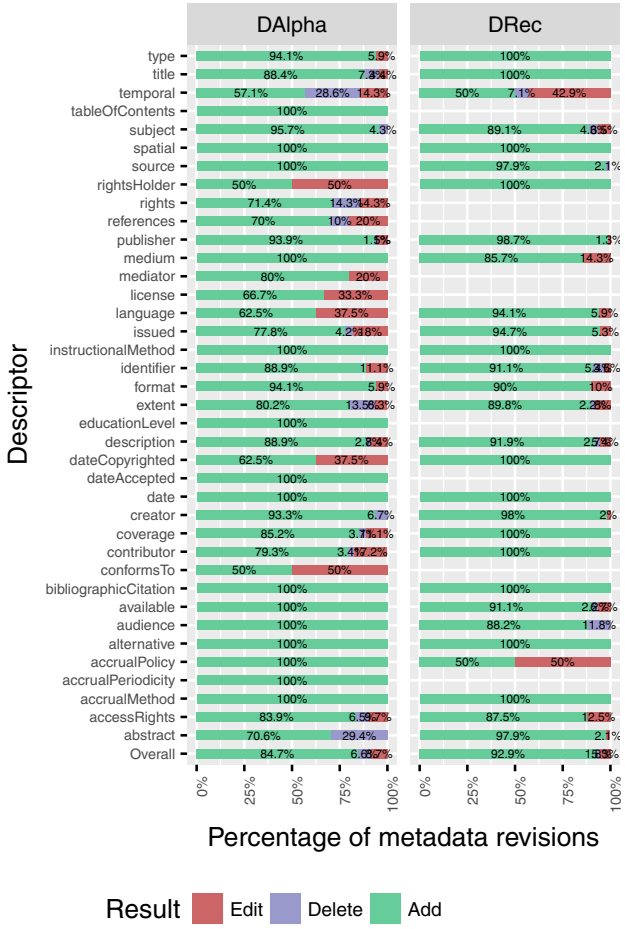

Fig. 10 Revision types per descriptor

siveness" 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 while ranking the descriptors—perhaps ending up having a slight impact in response times and originating occasional 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). 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 toward D_{Rec}, with a superior average as well as an inferior standard deviation. Finally, the user satisfaction 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, plus 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 analyzed parameters.

7 Conclusions

With the strong push toward 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 support 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 relevant. In this experiment with usage-driven descriptor ranking, Dendro is configured to use only Dublin Core Terms descriptors, taking advantage of the fact that DC is arguably the most widely known and used schema in digital repositories.

Any attempt to improve the detail and comprehensiveness of metadata records by offering the researchers a broader 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 prompted 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. The main goal is to build a formula that can evolve according to evidence collected from user interactions, dataset contents and other sources of evidence on the relevance of descriptors for users or projects. User interaction is the source of evidence in the current version, and the formulas are linear combinations of metrics for features extracted from interactions, with weights and bounds set empirically according to the observation of

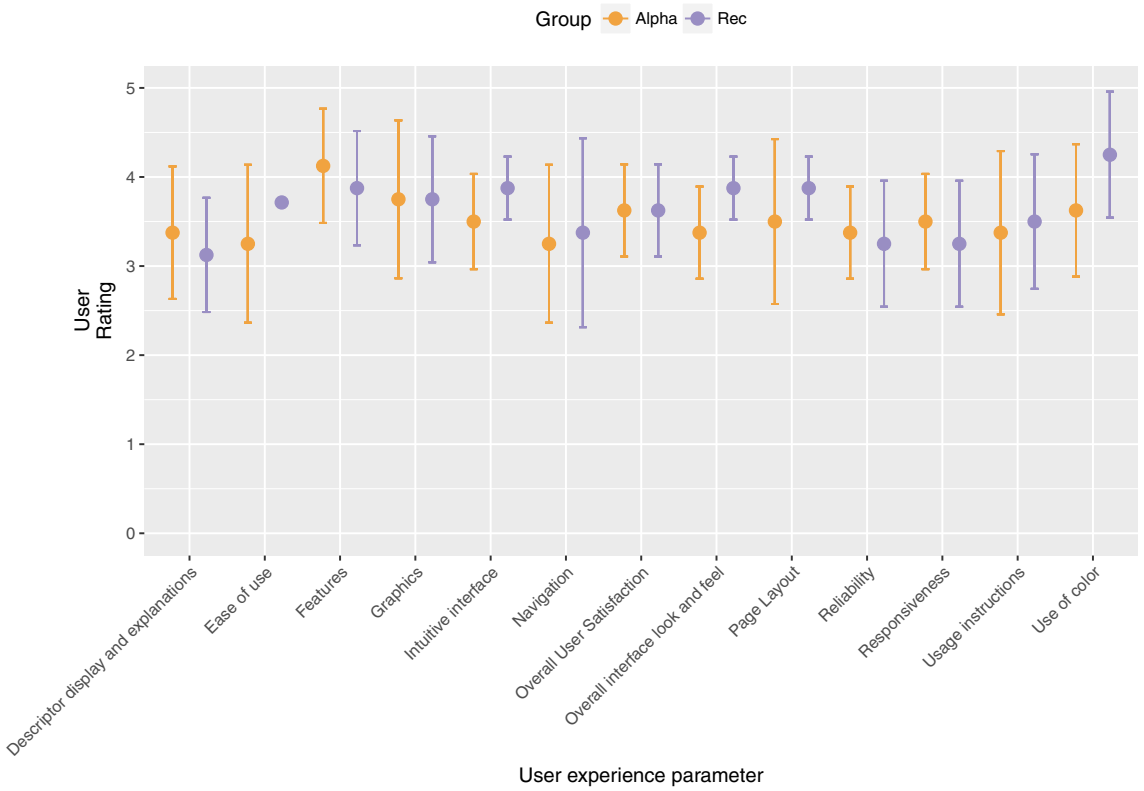

Fig. 11 User ratings on the various surveyed parameters

typical sessions. To capture user interactions, Dendro was extended to log them and they were classified in several types.

The ability of Dendro to record the full history of metadata changes in any record made it possible to analyze the evolution of metadata records. We used this to show that the introduction of descriptor ranking features can help reduce user mistakes when filling in metadata, as shown by a lower percentage of descriptor corrections and deletions when descriptor ranking is present. We believe that such results may add to existing work that reports on the curation effort required to ensure the quality of user-submitted datasets containing DC-compliant metadata [18].

Our users were asked to participate in a study over 3 weeks, during which they used 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 files and folders.

To evaluate the response of the users to the platform, we carried out a user satisfaction survey. The results are positive (average rating of 3 and above) across all analyzed user experience parameters, which range from reliability to the overall look and feel of the platform and its user-friendliness. The open-answer section of the questionnaires highlighted a few design flaws of the Dendro solution; most were minor bugs that, while "somewhat annoying" in the words of our users, did not prevent them from carrying out their tasks.

The base data of these studies are available in a demonstration project at our Dendro demonstration instance[10] and Dendro is under active development as an open-source project on GitHub[11].

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 the first user test of Dendro. It was also a load test of the platform, since all participants 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**
   After an experiment like this is set in motion, there can be

[10] http://dendro.fe.up.pt/demo.

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

no further modifications to the code running on the test instances, so the deployment has to be carefully executed. While 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. One cannot underestimate the importance of writing quality software to support user studies.

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 participants 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 welcome 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. 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 in our case provided a realistic data description experience but to a certain degree introduced additional variables in the experiment. An example is the button for switching between the "Automatic" and "Manual" metadata editor mode. 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 the responses 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 experimental scenarios**
   The design of the experiment can also be improved. While it is true that splitting the user sample *apriori* ensures the absence of learning bias, in our case the resulting subgroups were 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 do not have yet. Second, the selection of features is based on our intuition regarding their quality as predictors of descriptor relevance. Third, we used a convenient sample of users, namely a group of students with information science background and some basic familiarity with Dublin Core.

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 the current user, or collaborators of the active project, to be on the list. As Dendro is used by more researchers we will be able to log more user interactions. When the volume of interactions becomes large enough for the application of machine learning approaches, we will explore *learning to rank* in our descriptor rankings. Learning to rank refers to the application of machine learning techniques to train the model used in a ranking task [30].

We are also planning to carry out additional experiments combining domain-specific descriptors from ontologies other than DCTERMS. 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, offloading curators from metadata creation to the tasks of validation and improvement of existing Application Profiles.

The logging system can be made more complete to allow us to study more variables. It will be interesting, for example, to record web page loading timestamps so that we can determine how much time it takes for users to select and fill in a descriptor after the list of descriptors is presented. This will allow us to examine the learning behavior of our users and compare their performance with and without descriptor ranking.

A user-level quantitative analysis will provide more insight on why, for example, the U_{Rec} subgroup ended up saving a 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 may have decided to perform a much more comprehensive description of their selected datasets.

The "auto-complete" search box was used several times by the users to fetch specific descriptors. It may prove inter-

esting to analyze the typing sequences of 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 (since the `rdf:label` and the `rdf:comment` property of the descriptor are both used when searching for descriptors).

The descriptor lists emerge from the actual usage of descriptors in different research groups. In the long term, we believe that this information can be helpful for the formalization of an Application Profile [19], 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 [33,35]. However, it may provide additional quantitative information regarding the changing needs of a community—which is an important driving force for the improvement of a DCAP [27].

The development of Dendro, which started in 2014, continues in the context of the TAIL project[12]. TAIL started in 2016, ends in May 2019, and is developing and testing workflows for managing multi-disciplinary data in the long tail of science. Tools such as Dendro and LabTablet [2] will be put into production as part of the research data management workflow at INESC TEC, the University of Porto and the Research Center for Biodiversity and Genetic Resources[13] (CIBIO). This workflow will implement two[14] Dendro instances for research dataset preparation, integrated with two[15] CKAN[16]-powered research data repositories. TAIL explores metadata schemas besides Dublin Core, and works closely with researchers to define them. 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 data.

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. Allinson, J., Johnston, P., Powell, A.: A Dublin Core Application Profile for Scholarly Works. Ariadne, 50, (2007). [Originating URL: http://www.ariadne.ac.uk/issue50/allinson-et-al/. Accessed 15 Jan 2018
2. Amorim, R., Castro, J., Rocha, J., Ribeiro, C.: Engaging Researchers in Data Management with LabTablet, an Electronic Laboratory Notebook, pp. 216–223. Springer International Publishing, Cham (2015)
3. Amorim, R., Castro, J., Rocha, J., Ribeiro, C.: A comparison of research data management platforms: architecture, flexible metadata and interoperability. Univ. Access Inf. Soc. **16**(4), 851–862 (2017)
4. Ball, A.: Scientific data application profile scoping study report. Technical report, UKOLN, University of Bath, Bath, UK, (2009)
5. Bechhofer, S., Buchan, I., Roure, D.D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D., Owen, S., Newman, D., Sufi, S., Goble, C.: Why linked data is not enough for scientists. Future Gen. Comput. Syst. **29**(2), 599–611 (2011)
6. Berners-Lee, T.: Linked Data—Design Issues. http://www.w3.org/DesignIssues/LinkedData.html, (2008). Accessed 15 Jan 2018
7. Bizer, C., Heath, T., Berners-Lee, T.: Linked data—the story so far. Special issue on linked data. Int. J. Semantic Web Inf. Syst. **5**(3), 1–22 (2009)
8. Borgman, C.L.: The conundrum of sharing research data. J. Am. Soc. Inform. Sci. Technol. **63**(6), 1059–1078 (2012)
9. Boyko, A., Kunze, J., California Digital Library, Littman, J., Madden, L., Library of Congress, Vargas, B.: The BagIt File Packaging Format (V0.97). https://tools.ietf.org/html/draft-kunze-bagit-06 (2012) Accessed 15 Jan 2018
10. Coyle, K., Baker, T.: Guidelines for Dublin Core Application Profiles. http://dublincore.org/documents/profile-guidelines/ (2009). Accessed 15 Jan 2018
11. Dublin Core Metadata Initiative. DCMI Metadata Terms. http://dublincore.org/documents/dcmi-terms (2012). Accessed 15 Jan 2018
12. Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.0: Reference Description. http://dublincore.org/documents/1998/09/dces/ (2012). Accessed 15 Jan 2018
13. European Commission, Directorate-General for Research & Innovation. H2020 Programme, Guidelines on FAIR Data Management in Horizon 2020, Version 3.0. Technical report, 26 July (2016)
14. Eynden, V.V.D., Corti, L., Bishop, L., Horton, L.: Managing and Sharing Data: A guide to good practice. UK Data Archive University of Essex Wivenhoe Park Colchester Essex CO4 3SQ, 3rd edition (2011)
15. Gormley, C., Tong, Z.: Elasticsearch: The Definitive Guide, 1st edn. O'Reilly Media, Inc., Sebastopol (2015)
16. Goy, A., Magro, D., Petrone, G., Picardi, C., Segnan, M.: Ontology-driven collaborative annotation in shared workspaces. Future Gen. Comput. Syst. **54**, 435–449 (2016)
17. Greenberg, J.: Metadata capital: raising awareness, exploring a new concept economics of knowledge organization systems. Bull. Assoc. Inf. Sci. Technol. **40**(4), 30–33 (2014)
18. Greenberg, J., Swauger, S., Feinstein, E.: Metadata capital in a data repository. Proc. Int. Conf. Dublin Core Metadata Appl. **2013**, 140–150 (2013)
19. Heery, R., Patel, M.: Application profiles: mixing and matching metadata schemas. Ariadne, 25, (2000). Originating URL: http://www.ariadne.ac.uk/issue25/app-profiles/. Accessed 15 Jan 2018
20. Heidorn, P.B.: Shedding light on the dark data in the long tail of science. Library Trends **57**(2), 280–299 (2008)
21. Hey, T., Tansley, S., Tolle, K.: The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, October (2009)
22. Hodson, S.: ADMIRAL: A Data Management Infrastructure for Research Activities in the Life sciences. Technical report, University of Oxford (2011)
23. Hu, R., Pu, P.: 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

[12] https://www.inesctec.pt/en/projects/tail-PC05170.

[13] https://cibio.up.pt.

[14] https://dendro.inesctec.pt & https://dendro-rdm.up.pt.

[15] https://rdm.inesctec.pt and https://ckan-rdm.up.pt.

[16] https://ckan.org.

24. International Organization for Standardization. Space data and information transfer systems—Open Archival Information System (OAIS)—Reference model. Standard ISO 14721:2012, Geneva, CH, September (2012)
25. Jahnke, L., Asher, A., Keralis, S.D.C.: The Problem of Data. Council on Library and Information Resources (2012). [Originating URL: http://www.clir.org/pubs/reports/pub154. Accessed 15 Jan 2018
26. Joachims, T., Granka, L., Pan, B.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 154—161 (2005)
27. Krause, E.M., Clary, E., Greenberg, J., Ogletree, A.: 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, 63–75 (2015)
28. Lecarpentier, D., Wittenburg, P., Elbers, W., Michelini, A., Kanso, R., Coveney, P., Baxter, R.: EUDAT: A New Cross-Disciplinary Data Infrastructure for Science. Int. J. Digit. Curation **8**(1), 279–287 (2013)
29. Leonelli, S., Spichtinger, D., Prainsack, B.: Sticks and carrots: encouraging open science at its source. Geo: Geogr. Environ. **2**(1), 12–16 (2015)
30. Li, H.: A short introduction to Learning to Rank. IEICE Trans. Inf. Syst. **E94–D**(10), 1854–1862 (2011)
31. Lord, P., Macdonald, A.: Data curation for e-Science in the UK: an audit to establish requirements for future curation and provision. Technical report, JISC (2003)
32. Lyon, L.: Dealing with Data: Roles, Rights, Responsibilities and Relationships. Technical report, UKOLN, University of Bath (2007)
33. Malta, M., Baptista, A.: State of the Art on Methodologies for the Development of a Metadata Application Profile, pp. 61–73. Springer, Berlin (2012)
34. Malta, M., Baptista, A.: A Method for the Development of Dublin Core Application Profiles (Me4DCAP V0.1): A Description. In: Proceedings of the International Conference on Dublin Core and Metadata Applications 2013, pp. 90–103 (2013)
35. Malta, M., Baptista, A.: A panoramic view on metadata application profiles of the last decade. Int. J. Metadata Semant. Ontol. **9**(1), 58–73 (2014)
36. Martinez-Uribe, L.: Using the Data Audit Framework: an Oxford case study. Technical report, Oxford Digital Repositories Steering Group, JISC (2009)
37. Martinez-Uribe, L., Macdonald, S.: 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)
38. Piwowar, H., Vision, T.: Data reuse and the open data citation advantage. PeerJ, 1:e175, (2013) Originating URL: https://doi.org/10.7717/peerj.175. Accessed 15 Jan 2018
39. Rocha, J.: Usage-driven Application Profile Generation Using Ontologies. Ph.D. thesis, Faculdade de Engenharia, Universidade do Porto, May (2016). Originating URL: http://hdl.handle.net/10216/83993. Accessed 15 Jan 2018
40. Rocha, J., Castro, C., Ribeiro, J., Lopes, J.: Dendro: Collaborative Research Data Management Built on Linked Open Data, pp. 483–487. Springer International Publishing, Cham (2014)
41. Rocha, J., Ribeiro, C., Correia Lopes, J.: Ontology-based multi-domain metadata for research data management using triple stores. In: Proceedings of the 18th International Database Engineering & Applications Symposium, IDEAS'14, pp. 105–114, New York, NY, USA, ACM (2014)
42. Rocha, J., Ribeiro, C., Correia Lopes, J.: The Dendro Research Data Management Platform: Applying Ontologies to Long-Term Preservation in a Collaborative Environment. In: Proceedings of the 11th International Conference on Digital Preservation, Ipres 2014, Melbourne, Australia, October 6–10, 2014 (2014)
43. Rocha, J., Ribeiro, C., Lopes, J.: Managing research data at U. Porto: requirements, technologies and services. Innovations in XML Applications and Metadata Management: Advancing Technologies, IGI Global:174–197 (2013)
44. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: 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)
45. Magazine, Science: Dealing with data. Challenges and opportunities. Introduction. Science (New York, N.Y.) **331**(6018), 692–693 (2011)
46. Silvello, G.: Theory and practice of data citation. J. Assoc. Inf. Sci. Technol. **69**(1), 6–20 (2018)
47. Sinha, R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: Proceedings of the DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries, volume 01/W03, Dublin City University, Ireland, 18–20 June (2001)
48. Sinha, R., Swearingen, K.: The role of transparency in recommender systems. In: CHI '02 Extended Abstracts on Human Factors in Computing Systems, CHI EA '02, pp. 830–831, New York, NY, USA, ACM (2002)
49. Strickroth, S., Pinkwart, N.: High quality recommendations for small communities: the case of a regional parent network. In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pp. 107–114, New York, NY, USA, ACM (2012)
50. Swanberg, S.: Inter-university consortium for political and social research (ICPSR). J. Med. Lib. Assoc. **105**(1), 106–107 (2017)
51. Swearingen, K., Sinha, R.: Beyond Algorithms: an HCI perspective on recommender systems. ACM SIGIR 2001 Workshop on Recommender Systems (2001), pp. 1–11 (2001)
52. The Data Seal of Approval Board. Implementation of the data seal of approval. https://assessment.datasealofapproval.org/assessment_114/seal/html/ (2014). Accessed 15 Jan 2018
53. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, **3**:160018 EP, 03 (2016)