FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Selection of important posts in the timeline of a social network

Miguel Guilherme Perestrelo Sampaio Pereira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Rocha da Silva, PhD

Second Supervisor: Maria Cristina de Carvalho Alves Ribeiro, PhD

24th July 2018

Selection of important posts in the timeline of a social network

Miguel Guilherme Perestrelo Sampaio Pereira

Mestrado Integrado em Engenharia Informática e Computação

24th July 2018

Abstract

Research data management aims to ensure that the whole process, from the collection of data to its dissemination and archiving is done in an organized and understandable way. This can be done through the use of descriptive names that clearly represent the files and what is contained therein, as well as a set of complete and suitable metadata. This guarantees that other researchers without previous knowledge of the dataset are able to easily understand it and reuse it in their research. Sharing research data is an important part of scientific research, as it allows the verification of results and enables others to ask new questions of existent data, facilitating the advance of the state of research and innovation.

Dendro is a collaborative research data management platform, currently under development at Faculdade de Engenharia da Universidade do Porto, that aims to establish a close proximity to the researchers, involving them in the management and description of their data in the early stages of research. It is supported by a fully open-source environment built for cloud-level scalability. Users can deposit and describe files of any type using descriptors such as creator and modification date, using a file manager similar to Dropbox.

Social Dendro, a social extension for Dendro, consists of a timeline that lists, in the form of posts, the editions/uploads of files and folders in the user's projects. These posts can be liked, commented and shared, allowing the researchers to be notified of the modifications as soon as possible, as well as giving feedback on them in a simple and practical way.

In the case of a user with a high number of projects or projects with many modifications, the timeline can easily become "flooded" with posts. This can lead to the user having to spend a lot of time scrolling the timeline to find relevant posts, which in turn can lead to the users not using the timeline as much as expected, reducing their participation in the data description.

To overcome the limitations of the timeline, this work proposes the design and implementation of a ranking method of the most relevant posts in the timeline, in a way the user can effortlessly understand the modifications that occurred since the last time he or she visited the platform. It will be necessary to find which of the post features make it more relevant and rank the posts based on them. Posts with a higher score will be shown first in the timeline.

This implementation is expected to promote data description, promoting the participation of the team members in the development of the project and making it more efficient for them to keep track of the other member's contributions.

Resumo

A gestão de dados de investigação tem como objetivo garantir que todo o processo, desde a recolha de dados até à sua disseminação e arquivo, é feito de uma maneira organizada e compreensível. Isto pode ser feito através do uso de nomes descritivos que representem claramente os ficheiros e o que está contido neles, bem como através de um conjunto de metadados completo e adequado. Assim, garante-se que outros investigadores sem conhecimento prévio do dataset consigam facilmente percebê-lo e reutilizá-lo nas suas investigações. Partilhar dados de investigação é uma parte importante da investigação científica, uma vez que permite a verificação dos resultados e o levantamento de novas questões sobre esses dados, facilitando o avanço do estado da pesquisa e da inovação.

O Dendro é uma plataforma colaborativa para a gestão de dados de investigação, que está atualmente a ser desenvolvida na Faculdade de Engenharia da Universidade do Porto, e que tem como objetivo estabelecer uma relação de proximidade com os investigadores, envolvendo-os na gestão e descrição dos seus dados na fase inicial da investigação. É suportado por um ambiente completamente *open-source* construído para ter uma escalabilidade ao nível da *cloud*. Os utilizadores podem depositar e descrever dados de qualquer tipo utilizando descritores tais como criador e data de modificação, através do uso de um gestor de ficheiros semelhante ao Dropbox.

O Social Dendro, a extensão social do Dendro, consiste numa *timeline* que lista, na forma de *posts*, as edições/*uploads* de ficheiros e pastas nos projetos do utilizador. Estes posts podem receber *likes*, comentários e ser partilhados, permitindo aos investigadores serem notificados das modificações o mais cedo possível, bem como dar *feedback* dessas mesmas modificações de uma maneira simples e prática.

No caso de um utilizador com um elevado número de projetos ou projetos com muitas modificações, a timeline facilmente fica "inundada"de *posts*. Isto pode levar a com que o utilizador tenha de gastar muito tempo a fazer *scroll* até encontrar os *posts* relevantes, o que, por sua vez, pode fazer com que o utilizador não utilize a *timeline* tanto como esperado, reduzindo a sua participação na descrição dos dados.

De maneira a ultrapassar as limitações da *timeline*, este trabalho propõe o desenho e implementação de um método de classificação dos posts mais relevantes na *timeline*, de maneira a que o utilizador possa, com pouco esforço, perceber as modificações que occorreram desde a sua última visita à plataforma. Será necessário perceber que características do *post* o tornam mais ou menos relevante e classificar o *post* com base nelas. *Posts* com uma classificação mais alta serão mostrados em primeiro lugar.

Com esta implementação espera-se promover a descrição dos dados, fazendo com que os membros da equipa participem mais no desenvolvimento do projeto e sejam capazes de estar a par das contribuções dos outros membros eficientemente.

Acknowledgements

First, I would like to thank my coordinator, João Rocha da Silva, and the researcher Nelson Pereira for always being available and ready to jump in and figure out the problem. This dissertation would not have been possible without their help.

To all my friends, for the support, for the laughs and for making these the best 5 years: thank you.

And finally, a huge thank you to my parents for always supporting me and because this would not have been possible without them.

Miguel Pereira

"Sometimes the questions are complicated and the answers are simple."

Dr. Seuss

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

CSV	Comma-separated values
HITS	Hyperlink-Induced Topic Search
HTML	HyperText Markup Language
ICT	Information and communication technology
LOD	Linked Open Data
ORM	Object-relational mapping
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
RDM	Research data management
SPARQL	SPARQL Protocol and RDF Query Language
TDD	Test Driven Development
UML	Unified Modeling Language
URI	Uniform Resource Identifier
XML	Extensible Markup Language

Chapter 1

Introduction

This chapter presents the context of the work (Sec. 1.1), as well as the motivation to deal with the described problem and what we aim to obtain with the work (Sec. 1.2). In the end, the structure of this dissertation is briefly explained (Sec. 1.3).

1.1 Context

The work falls within the scope of the research data management, with special focus to the curation and preservation of research data. The proposed solution will build on the open-source platform Dendro [RARC14], currently being developed at Faculdade de Engenharia da Universidade do Porto.

1.2 Motivation

With the production of research data quickly increasing, it becomes more and more crucial to find the means to properly manage these data [JAK12]. But many times, research institutions in the "long tail" of science don't have the financial resources to support the curation and description of the data produced by their researchers [P. 08]. Researchers are the most knowledgeable about their data, so involving them in the data management process is one of the possible solutions to this problem. The introduction of social-network concepts to data management platforms enables researchers to participate in the data description and promotes the communication between team members. The further refinement of these concepts and techniques will help researchers even more in the description of their data.

1.3 Dissertation Structure

Besides this introduction, the dissertation has 7 more chapters. Chapter 2 presents a brief description of research data management (RDM) and introduces the concepts of Linked Open Data and ontologies and how they can be used in RDM, ending with a look at the application of social-network techniques in RDM. Chapter 3 introduces Dendro and its social extension Social Dendro. Chapter 4 gives an overview of the existing post ranking methods, making a comparison between them. Chapter 5 explains the problem that motivated the work and describes the implemented solution. Chapter 6 gives a high-level overview of the implementation. Chapter 7 presents the evaluation method used to properly evaluate the quality of the proposed solution and explains the results. Finally, chapter 8 presents the final balance, as well as the possible future work.

Chapter 2

Research Data Management

This chapter presents a brief description of research data management, its aim and its importance (Sec. 2.1); an explanation of the research data lifecyle (Sec. 2.2); a brief overview of Linked Open Data and how it can be used in research data management (Sec. 2.3); and finally, a look at the application of social-network techniques in research data management and how it can improve the process (Sec. 2.4).

2.1 Introduction

Data is more than numbers. It may take many forms, both physical and digital: notebooks, video recordings, sound, images, games, algorithms. Data is processed, analyzed, combined; data has a story, it is neither static nor isolated [SR15]. During the course of research, a great amount of time and labor is spent by investigators and their associates compiling, managing and interpreting data and publishing results—data can be considered the "lifeblood of research" [Bor12], so good practices for managing data should be one of the main priorities of researchers.

Research data management aims to ensure that the whole process, from data collection to the dissemination and archiving of results, is "organized, understandable and transparent" [SR15]. Types of data management include the use of descriptive names for variables, files and folders that make it clear what they represent or what is contained therein, unique identifiers for study participants and study workflows describing the analysis methodology [SR15]. These practices allow other researchers without previous knowledge of the dataset to easily understand and reuse it, as the meanings, definitions and relations of data are clearly defined.

Sharing data is an important part of the scientific research as it allows to reproduce or to verify research, makes the results of publicly funded research available to the public, enables others to ask new questions of existent data and facilitates the advancing of the state of research and innovation [Bor12, WJ11]. Not only the general research community benefits from shared data, as it has been

shown that publications whose data has been made publicly available yield an higher citation rate [PDF07]. The citation rate is often used by many investigators as a currency of value [PDF07].

2.2 Research data lifecycle

Research data have a life beyond the investigation that produces them. Researchers may carry on working on data after the project is over, subsequent projects may interpret and supplement the data, and other researchers may also re-use the data. The concept of research data lifecycle identifies the stages through which research data is collected, recorded, processed and results are published in order to guarantee successful management and preservation of data for use and reuse.

There are multiple versions of the research data lifecycle, one of them being the DCC Curation Lifecycle Model shown in Figure 2.1. This model provides a "graphical high-level overview of the stages required for successful curation and preservation of data from initial conceptualization or acquisition" [Hig08], and it can be used in conjunction with other models and frameworks to help plan activities at more granular levels. As the needs of users may vary, they can enter at any stage of the lifecycle.

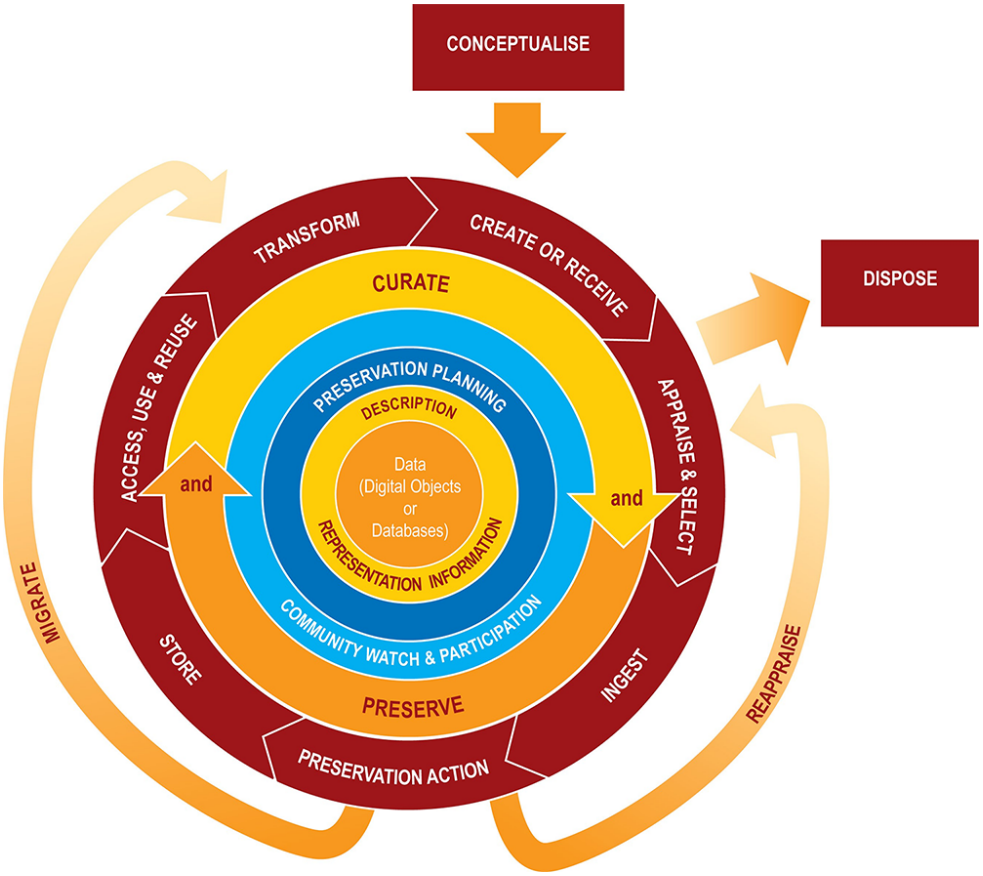

Figure 2.1: The DCC Curation Lifecycle Model [Hig08]

Data, comprised of digital objects and databases, is at the center of the lifecycle. Moving outward, there are the full lifecycle actions: description and representation information refers to the use of appropriate metadata and representation information required to understand the material; preservation planning includes the plans for management and administration of all curation lifecycle actions; community watch and participation means following the community activities and participating in the development of shared standards and tools; curate and preserve, by undertaking the steps planned to bolster curation and preservation during the whole of the lifecycle [Hig08].

After, we have the sequential actions: conceptualize, conceive and plan the creation of data, including the capture method and storage options; create and receive data; appraise and select refers to the evaluation and selection of data for long-term curation and preservation; ingest, meaning transfer data to an archive, repository, data center or other custodian; preservation action is the stage where actions are undertaken to ensure long-term preservation and retention of the authoritative nature of data, ensuring the data remains authentic, reliable and usable while maintaining its integrity; store data in a secure manner; access, use and reuse, ensuring the data is accessible to both designated users and re-users, on a day-to-day basis; transform, create new data from the original [Hig08].

Finally, the occasional actions: dispose of data which has not been selected for long-term curation and preservation, by transferring it to another archive, repository data center or destroying it; reappraise, meaning returning data which fails validation procedures for further appraisal and re-selection; migrate data to a different format, which may be done to accord with the storage environment or to ensure the data's immunity from hardware or software obsolescence [Hig08].

There are simpler models like the one proposed by the UK Data Service[2], presented in Figure 2.2, which includes seven sequential stages: planning research, data management, processing protocols; collecting data and capturing data with metadata; processing and analyzing data, which includes transcribing, validating, cleaning, deriving, documenting, interpreting, producing outputs; publishing and sharing data, by first establishing copyrights and creating user documentation and discovery metadata; preserving data, which can encompass migrating data to the best format/media, curating; re-using data in follow-up research or just to conduct a secondary analysis or scrutinize findings.

Both models are pretty similar and suggest the same actions, the main difference being that the DCC Curation Lifecycle Model suggests the curation and preservation actions as continuous actions in the lifecycle, instead of just stages of the process.

2.3 Linked Open Data in research data management

On the original Web, there was only interchange of documents. Search engines index the documents and analyze the link structure between them to infer potential relevance to users' queries

[1]Link: https://www.ukdataservice.ac.uk/manage-data/lifecycle
[2]Link: https://www.ukdataservice.ac.uk/

Figure 2.2: Research data lifecycle[1]

[BP98], while hypertext links allow users to navigate through them using Web browsers.

Up till recently, the standards that have allowed this "Web of documents" to prosper have not been applied to data. Often, data is published on the Web in formats such CSV, XML or HTML tables, which considerably sacrifices its structure and semantics. But the adoption of the Linked Data principles in recent years has brought to the expansion of the Web with a "global data space connecting data between different data sources" [BHBL09] (people, books, music, scientific data, reviews, etc.)—the Web of Data.

Linked Data refers to "data published on the Web in such a way that it is machine-readable" [BHBL09], its definition is explicitly defined, as well as the relationships between the data. Linked Data allows you to find other data, related to the data you already have. To realize this Web of Data, in 2006, Berners-Lee delineated a series of standards for publishing data on the Web [BL06]. These rules refer to the use of Uniform Resource Identifiers (URIs) and the Resource Description Framework (RDF) model. The RDF model encodes data in the form of <subject, predicate, object> triples. The subject and the predicate are both URIs, while the object can be a URI or a string literal. This provides a generic, graph-based data model with which to structure and link data that describes things in the world [BHBL09]. But different systems often make use of different concepts and terminology and RDF is not enough to describe the sometimes complex relationships between data—the use of ontologies becomes imperative. An ontology is a "shared understanding of some domain of interest" [UG96] to which we can call vocabulary. These vocabularies describe sets of classes and properties used to characterize entities and how they related to each other. Some of the tools used to conceive these vocabularies are the RDF Schema (RDFS) and the Web Ontology Language (OWL). RDFS provides a "data-modelling vocabulary for RDF data" [GB14] while the OWL is an "ontology language for the Semantic Web with formally defined meaning" [PSRP+12].

Evidence of the emergence of the Web of Data and perhaps the most visible example of adoption and application of the Linked Data principles has been the Linking Open Data project[3]. Linked Open Data (LOD) consists in converting, according to the Linked Data principles, data sets that are available under open licenses to RDF, publishing them on the Web and interlinking them with each other [BHBL09, BHIBL08]. Other example of the use of LOD is the DBpedia project, that "builds a large-scale, multilingual knowledge base by extracting structured data from Wikipedia editions in 111 languages" that can be used to answer expressive queries [Biz12].

The same Linked Data principles can be applied in research data management. By using a data model based on LOD, metadata values gain structure and explicit meaning, allowing datasets, papers, researchers and other research-related resources to be connected by meaningful links. This gives way to a simple and flexible data model that is prepared for obsolescence, as the data can be exported as LOD [dSCRL14]. This concept is already being implemented in some platforms such as the Dendro platform [dSCRL14] and B2NOTE [KCdS+17].

[3]Link: https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

2.4 Social networks in research data management

Nowadays, information and communication technology (ICT) services play a central role in the research activities. Besides supporting day-to-day data gathering (computers, connection to data centers, services for data management, processing), ICT services are also used for publishing and reusing the resulting research products [ACC+15]. The benefits of publishing research results go from the better verification and evaluation of scientific results to the maximization of research re-use, reducing the costs of research [ACC+15].

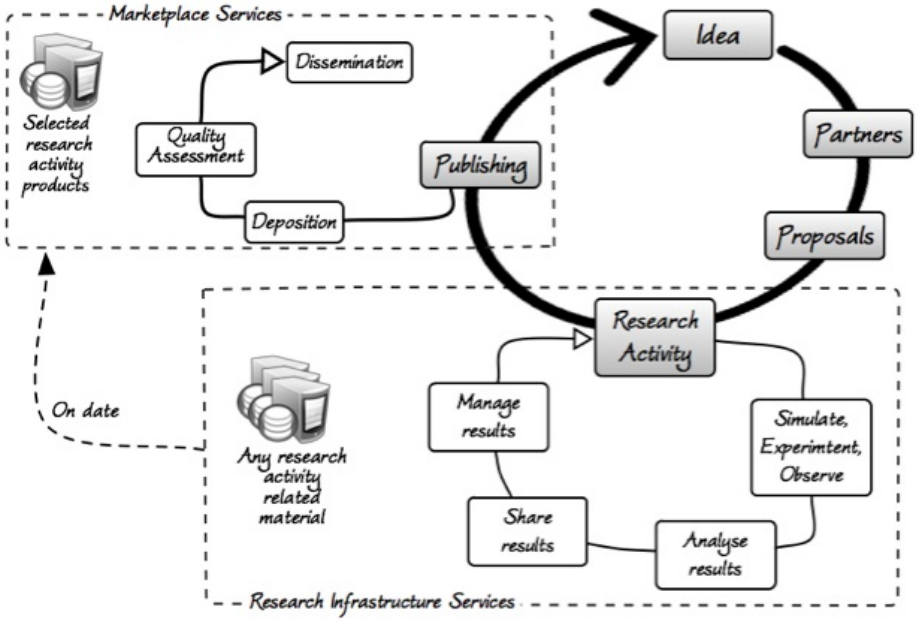

Figure 2.3: Publishing in a Research Lifecycle [ACC+15]

Figure 2.3 shows a model in which publishing represents the final action of the research activity lifecycle, where the product is deposited into accredited repositories to be submitted to a peer-review process and then disseminated. Research activities are conducted in the research infrastructures, while publishing takes places "elsewhere", only when the researcher believes the publication is mature enough. But this leads to some drawbacks that limit the effective interpretation of research results and their subsequent evaluation and reuse [ACC+15]:

- The published products are deprived of any relationship to the original research activity (decontextualisation) and are frozen to their publishing status (staticity);
- Ineffective peer-review, as other than papers, proper evaluation can only be done in the scope of the research activity;
- Products are scattered across several marketplace repositories (fragmentation), and there is no guarantee that the products contain relationships between them.

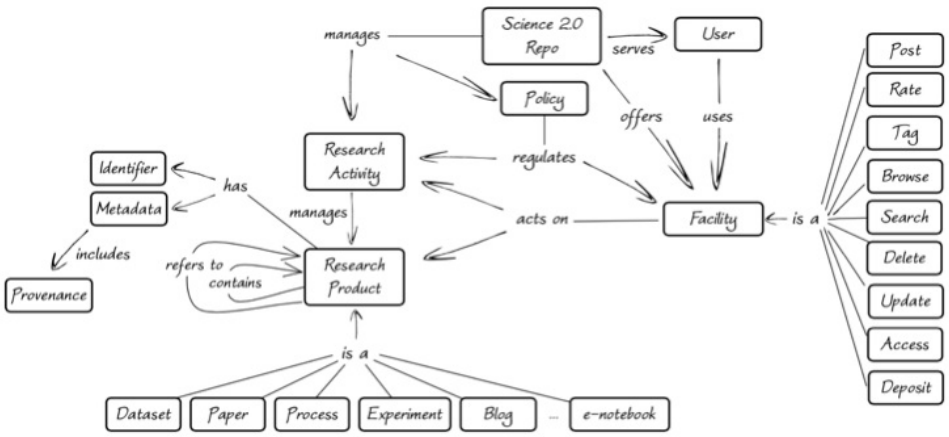

Figure 2.4: Science 2.0 Repositories Conceptual Model [ACC+15]

Due to the implicit drawbacks of publishing, some products end up by never being published.

As researchers are the main responsible for their data, involving them in the data management process leads to higher quality research data. But the current publishing model and its drawbacks deviate researchers from this process. The Science 2.0 Repositories concept introduces an idea to overcome this problem: publishing "within" the research infrastructure and "during" the research activities, as apposed to publishing "elsewhere" and only at the mature stage. It also introduces social networking practices, as way to modernize scientific communication, by posting rather than deposition, using "likes" and "open discussion" for quality assessment, sharing rather than dissemination [ACC+15]. It idealizes a single service offering two kinds of end-user functionalities: repository-oriented facilities and collaboration-oriented facilities. Figure 2.4 presents a basic model of the Science 2.0 Repositories (SciRepos). In this model, any output of the research process is potentially a relevant research product (in the form of datasets, papers, experiments and others), as such each of them is subject to the publishing stages (deposit, quality assessment, dissemination) and is provided with comprehensive metadata automatically generated via the hooking facility. SciRepos offer repository-oriented (Deposit, Access, Update, Delete, Search, Browse) and collaboration-oriented (Post, Rate, Tag) functionalities [ACC+15].

2.5 Summary

Research data management plays a big role in the generation of quality research data, and the involvement of researchers in this process is fundamental, as they are the most knowledgeable about their data. The DCC Curation Lifecycle Model suggests that curation and preservation should be done throughout the entirety of the investigation activities and by using innovative technologies like Linked Open Data and applying concepts of social networking to research data management,

it is possible to create a simple and flexible model that engages researchers in the data management process in the early stages of the investigation, which leads to quality metadata and as consequence, quality research data. Dendro [dSCRL14], and its social extension Social Dendro [PdSR17] already implement some of these functionalities.

Chapter 3

Dendro Platform

This chapter introduces Dendro and its main functionalities (Sec. 3.1), as well as a brief technological overview (Sec. 3.2); then continues to presenting Social Dendro (Sec 3.3) and ends with a short summary of the chapter (Sec. 3.4).

3.1 Introduction

Dendro is a collaborative research data management platform, currently under development at Faculdade de Engenharia da Universidade do Porto, that aims to establish a close proximity to the researchers, involving them in the management and description of their data in the early stages of research [dSCRL14]. Research groups have the ability to create projects to then store, describe and share their data, using a file manager similar to the one provided by Dropbox[1].

Dendro is not a long-term preservation platform, acting only as staging area for data curation. The target users are researchers on the so-called "long tail" of science that don't have the financial resources to properly curate and describe their data [P. 08]. Its user-friendly interface, focusing on incremental data description, and no requirements for mandatory metadata, make it stand out from other similar platforms.

The ontology-based approach of Dendro allows researchers to describe their data, by allowing domain-specific ontologies to be used in resource description. This guarantees Dendro is capable of meeting the needs of the different research groups. There is also a recommender system for descriptors that assists users in describing their data.

3.2 Technological overview

Dendro allows users to describe their files and folders by using freely combined descriptors from ontologies previously loaded into the platform. These ontologies can be modeled using high-

[1] https://www.dropbox.com/

level tools like Protégé and then be shared on the web to be reused by others. Resources (files and folders) can be versioned by means of archived copies of the edited resource. Dendro also provides a free-text search functionality powered by an ElasticSearch index.

The use of ontologies is supported by the triple-based data model of Dendro. As a consequence, Dendro users are building a linked data knowledge base as they interact with the platform. This simple and flexible ontology-based data model assures that data is prepared to survive the obsolescence of the platform, as it becomes self-documented and can be fully exported in RDF.

Data access in Dendro is performed by queries written in SPARQL, which is executed by an instance of OpenLink Virtuoso. Queries in SPARQL are much more simple and expressive when compared with a relational approach, as previous knowledge of the database schema is not required [dSRL14, RARC14].

3.3 Social Dendro

Social Dendro, the social extension for Dendro, takes from the vision presented in Science 2.0 Repositories [ACC+15], that advocates that the deposit of data should be made as soon as possible in the research lifecycle, and implements some social networking concepts as a way to better support the research data management [PdSR17]. As continued collaboration between researchers in a project is a big step for successful data management, applying these techniques might favor communication between team members, as well as give a better understanding of the contributions of each member.

Social Dendro consists of a chronological timeline of posts, each representing the addition, edition and removal of a file or folder in a project or the edition of a file's metadata. This allows project members to have a clear view of the ongoing activities of research. Posts can be liked, commented and shared, which works as an informal peer-review system. This allows the research team to have feedback on new developments as soon as possible, as well as motivate researchers to be involved in the data management process.

3.4 Summary

With its innovative ontology-based approach and user-friendly interface, Dendro aims to promote data description in the early stages of research, especially among users without data management knowledge. Social Dendro builds on that by adding functionalities that further promote and facilitate communication between project members, helping produce better data.

Chapter 4

Post Ranking

In this chapter, an overview of the existing methods to post ranking is provided, by comparison of several different approaches (Sec. 4.1); in the end, a short summary is written, describing the results of this research (Sec. 4.2).

4.1 Related work

In order to get a sense of the existing solutions for post ranking within a social network, 7 articles presenting and describing methods to rank posts in a social timeline were analyzed.

More than half the solutions based their evaluation of the post on what was written on it and on how other users reacted to it, be it likes, comments or shares. Another frequent approach in the analyzed methods was the use of some form of probabilities, either the probability that the post is interesting to the user it is shown to, or the probability of that user interacting with the post in the form of likes, comments or shares.

Table 4.1 shows a comparison of different post ranking approaches, based on a set of 9 parameters:

- **Relevance filtering** — The method filters the posts, showing only the most relevant ones. The relevance of a post can be in regard to a specific query or topic, or just general relevance to the user;
- **Relevance ordering** — The method orders all the posts based on a relevance indicator, showing the complete timeline as opposed to filtering that generates a partial timeline.
- **Redundancy avoidance** — The method takes into consideration possible duplicate posts, by evaluating the content of the posts, as to not show redundant information in the timeline;
- **Semantic analysis** — The method semantically analyses the content of the post. This can be used to avoid redundancy and to evaluate the relevance of a post, given a topic or query;

- **Analysis of author's behavioral data** — The method takes into consideration the author's previous actions. For example, past publications and how the generality of users reacted to them;
- **Analysis of user's behavioral data** — The method takes into consideration the user's previous actions. For example, which posts he or she usually likes or comments, in order to get a term of comparison between posts;
- **Analysis of users' engagement** — The method evaluates how other users reacted to the post, that is the comments, likes and shares the post has;
- **Analysis of post features** — The method takes into consideration post features such as text length and publication date;
- **Use of probabilities** — The method uses some form of probabilities when calculating the post's score.

These parameters were chosen after an evaluation of each technique, as a way of understanding what each of them took into consideration when ranking a post, so that it was possible to find some common ground for a successful comparison. The methods are classified as having ("yes") or not having ("-") each of these features.

Table 4.1: Comparison of different approaches

	[FFYZ16]	[ATD17]	[WV14]	[YLLR12]	[YFFZ16]	[CCAB12]	[Bac16]
Relevance ordering	-	-	-	Yes	-	Yes	Yes
Relevance filtering	Yes	Yes	-	-	Yes	-	-
Avoid redundancy	Yes	Yes	-	-	Yes	-	-
Semantic analysis	Yes	Yes	-	-	Yes	-	Yes
Analysis of author's behavioral data	-	-	-	Yes	-	Yes	-
Analysis of user's behavioral data	-	-	Yes	-	-	Yes	Yes
Analysis of users' engagement	-	Yes	Yes	Yes	-	Yes	Yes
Analysis of post's features	-	-	Type, length	-	-	Age of tweet	-
Use of probabilities	Yes	-	Yes	-	-	Yes	Yes

Fan *et al.* (2016) proposed an adaptive evolutionary filtering algorithm, integrated with a hierarchical tweet representation model, to filter interesting tweets from the twitter stream with respect to user interest profiles, and a maximal relevance model in fixed time window to measure the overall quality of a potential tweet [FFYZ16]. Given a query from the user interest's profile which reflects the general topic the user cares, this algorithm monitors the real-time twitter stream to filter tweets that may be of interest to the user and then pushes them to the user as soon as possible, constructing an evolutionary timeline which consists of the pushed tweets, chronologically ordered.

When there is a new tweet, the relevance between the tweet and the query reflecting the user's interests is estimated, using a language modeling approach that estimates the word distribution of both the tweet and the query. The relevance score of the tweet can then be computed by a negative Kullback–Leibler (KL) divergence function. If the relevance of a tweet is less than an appropriate threshold, it is discarded.

Having filtered a set of tweets in a fixed time window, the marginal relevance score of each one is calculated, with account to its relevance score and a diversity score considering the previously pushed tweets, as to avoid redundancy. Then, an overall constraint is employed to determine whether the quality of the tweet is high enough to be pushed to the user.

As opposed to the other studied techniques, the work done by Alonso *et al.* (2017) doesn't generate a collection of tweets, but a timeline from a popular topic based on the link's article title that is included in a tweet [ATD17].

Given a hashtag or entity, the algorithm produces a contextual vector representing a ranked list of n-grams for a set of tweets related to that hashtag. To do this, the tweets related to the particular hashtag are first aggregated and then a series of steps such as filtering by tweet quality score and removal of duplicates are performed. Then, the link titles are extracted from the tweets and a score for each is computed by measuring the similarity between the hashtag and the title, using the produced contextual vector, and multiplying this value by a counter that quantifies user engagement via retweets, likes, or shares. The document titles are ranked by this score.

Instead of filtering or ordering, Waldner and Vassileva (2014) used a different approach, by simply changing the way tweets are visually presented to the user, altering the position and color saturation of tweets based on their relevance [WV14]. To calculate the recommendation score, this method uses a recommender that is trained beforehand by asking the user to rate a set of thirty individual tweets by classifying them as interesting or uninteresting. To predict whether a tweet is interesting to the user, the recommender uses a Naives Bayes classifier, that is retrained every ten minutes using features from the rated tweets. The recommendation score is given by the probability of the tweet being interesting, divided by the sum of the probability of the tweet being interesting and the probability of the tweet being uninteresting. It is not specified exactly how these probabilities are calculated, only that features like author, hashtags and number of retweets and favorites are taken into consideration.

Yang *et al.* (2012) states that a considerable amount of the millions of posts generated in social networking services are mundane posts that are of interest only to the authors and possibly their friends. Considering that, the proposed method automatically discovers valuable posts that may be of potential interest to a wider audience, using a variant of the Hyperlink-Induced Topic Search (HITS) algorithm [YLLR12].

The Twitter structure is modeled as a directed graph, with the users and tweets being considered as nodes and the retweet relations between these nodes as directional edges. The intuition is that the authority of a user is important to infer the score of the tweets that the user published, so the HITS algorithm is run on the user side first. After that, the authority and hub scores of the tweet nodes are computed. In each iteration, each tweet node starts by inheriting the authority and hub scores of the user who published the tweet. The algorithm takes into account the number of retweets and also if the user who retweeted is a follower of the creator of the tweet or not.

When searching a given topic in Twitter, users are overloaded with many posts that lack any meaningful information. To overcome this, Yao *et al.* (2016) proposed a method to generate an informative, meaningful timeline by selecting a small set of representative tweets, using a framework that models the relevance, novelty and coverage of the tweet timeline [YFFZ16].

Given a query, an expansion language model is built, based on the original word distribution, and is used to compare the relevance between the query and the new tweet. If the tweet is relevant enough, it is added to the most similar cluster, or a new one can be started, if no cluster is similar enough. After that, the coverage property, reflecting the proportion of the information contained in the cluster covered by tweet, is determined. Finally, and using a variation of the TextRank algorithm [MT04], the score for each tweet is calculated by jointly modeling the relevance and novelty and interpolating the coverage score.

In the work done by Comarela *et al.* (2012), they first characterize a very large Twitter data set in order to understand user behavior and identify factors that influence user response or retweet probability. And then, show that some of these factors can be used to improve the presentation order of tweets to the user. Two different methods are presented: a Naive Bayes predictor that combines the empirically observed conditional response probabilities, and a Support Vector Machine classifier [CCAB12].

Both methods take into account the state of the user regarding their interaction with Twitter: online or offline. The timeline reorganization occurs on every change of state and on every tweet arrival. In practical terms, it can be very expensive to reorder every time a new tweet is posted, so this can be replaced by some time out mechanism or the arrival of at least a given number of tweets. By using this approach, the score of a tweet can change over time, once the reorganization can be performed several times during the time the user is online. A tweet which is interesting now may not be in the future. To determine the score, a set of three attributes are considered: the

age of the tweet, the average sending rate of tweets of the user that has sent and a binary indicator that indicates if the user has interacted with the sender of the tweet before.

The first method uses a Naive Bayes predictor to assign scores to tweets, that represent the probability that a user will interact with a tweet given a set of its attributes. The final score is given by the probability of a user interacting with a tweet, knowing the age of the tweet, the sending rate of the author and if there was any previous interaction between the users (the events are considered independent).

For the Support Vector Machine approach, a binary supervised classifier was used. First, considering the same attributes used in the first method, tweets are classified as interesting (most likely to interact with) and non interesting. Following that, the tweets are presented to the user: first the ones classified as interesting, in reverse chronological order, and after, all classified as non interesting, also in reverse chronological order.

Backstrom (2016) presented an overview of the methods used by Facebook to present multiple personalized news feeds, with the objective of increasing user interaction with its feed [Bac16].

Every time a user visits the feed, only the new content is ranked and put at the top, as to not keep showing the same stories every time. A user refreshing the feed many times will get an almost chronologically ordered feed. Anything the user hasn't seen is considered new to him: new friend sharing the same link he's seen, unseen old stories, seen stories with new comments.

Using machine learning, and taking into consideration things like the relationships between users and the number of likes and shares of a feed story, the probabilities of the user, clicking, liking, commenting, sharing, hiding that story are determined and aggregated in a probabilities vector that is then transformed in a single score. Different events have different weights, according to their significance. Information from surveys is also used in the scoring process.

In a first approach, decision trees with gradient boosting and logistic regression techniques were used to determine the scores. At the time of the conference, the method used consisted in training logistic regression jointly on decision trees, deep neural networks and sparse features. This performs better than any of the models on their own, allows for near real-time updates and is computationally cheap.

4.2 Summary

Of the 7 studied methods, 3 of them generate a timeline based on a specific topic or event, by filtering tweets/posts and showing only the more relevant or representative ones; other three compute scores for the whole timeline and order posts based on that score; and one doesn't filter or order, simply changes the appearance (color, size) of the tweets based on their relevance score.

User engagement (likes, comments, shares) is the parameter more methods (5) take into consideration when calculating the score, while semantic analysis and the use of probabilities are also popular in the different approaches (used in 4 of them).

Finally, some of the methods (3) analyze the user's behavioral data (previous actions) and 2 of them analyze the author's behavioral data (past publications).

Chapter 5

A post ranking approach for Social Dendro

This chapter explains the problem that motivated the work (Sec. 5.1), describes the implemented algorithm (Sec. 5.2), explains the used formula and the features taken into consideration (Sec. 5.3) and presents an example of the application of the ranking method (Sec. 5.4).

5.1 Problem

Social Dendro implements a timeline consisting of posts that represent and describe the modifications to the data subject to the curation process in the context of a project. For a single user, its timeline will consist of a chronological listing of all the actions over the files metadata (addition, edition and removal), as well as the addition, edition and removal of files and folders in the projects the user participates [PdSR17]. These posts can be liked, commented and shared allowing the members of the research team working on the Dendro project to be notified of the modifications as soon as possible, as well as to give feedback on them in a simple and practical way.

In the case of a user with a high number of projects or projects with many modifications, the timeline can easily become "flooded" with posts, which can make it harder to distinguish between the most and least relevant posts. This can lead to the user having to scroll for longer periods of time in order to find relevant posts, which in turn can lead to a diminished interaction with the timeline, reducing the user participation in the data description, while the excess of non-relevant information can hinder collaboration between team members.

Figure 5.1 shows an example of a timeline, chronologically ordered, with several different types of posts, each type marked with a different pattern. The types of posts are creation (4) and deletion (1) of directories, upload (3) and deletion (2) of files, metadata changes (5) and manual posts (6). Just a small number of actions on a project can generate a timeline that can easily seem chaotic to the users, not allowing them to get a clear view of the activities of the projects they

contribute to. For example, uploading a couple of files and then creating some folders to organize the files would create a number of posts without much interest.

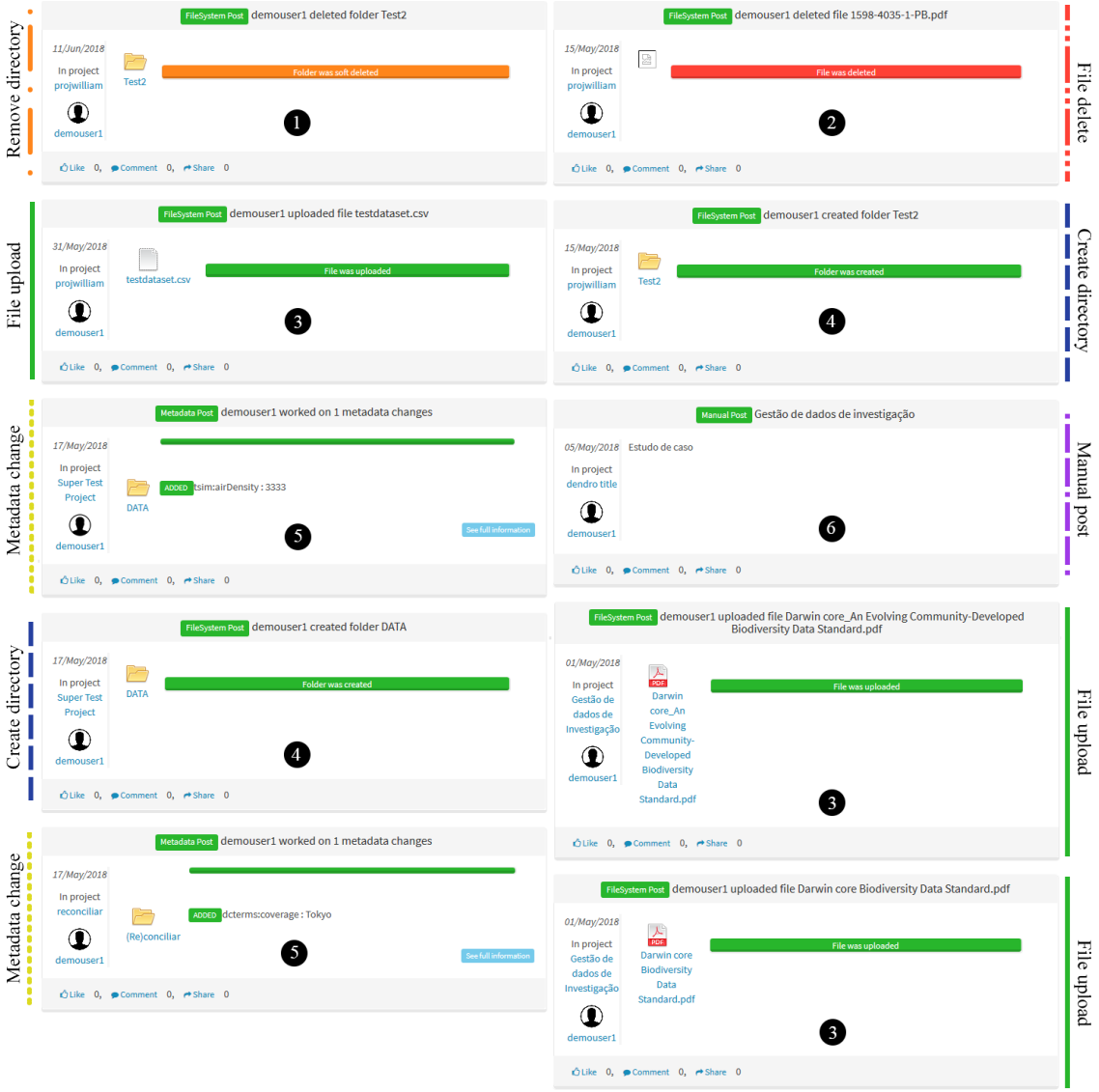

Figure 5.1: Chronologically ordered timeline

5.2 Solution description

The solution was devised based on the approaches of the existing post ranking methods (Ch. 4) and on the particularities of the Social Dendro timeline, as well as taking into consideration the main goal of the timeline—keeping researchers up to date on their projects while promoting data description.

The solution is characterized as follows:

Relevance ordering

In order to guarantee researchers are shown all the information they may care about, we opted for a relevance ordering approach. This guarantees all posts are shown in the timeline, so researchers are able to see all the actions on their projects.

Rank new/modified posts

So as to not overload the system scoring and ranking all posts every time the user accesses the timeline, we decided to rank only new posts, meaning all the posts made since the user's last visit and posts with new comments. Every time the user visits the timeline, the new content will be ranked and put at the top. This also prevents always putting the same content at the top and keeps an understandable model of things flowing down the page. Figure 5.2 shows an example of this approach: when the user visits the timeline at 1:00 PM, 3 new posts will be presented to the user, ordered by their relevance score; when the user visits the timeline one hour later, there will be 2 new posts; even though these posts have a lesser score than some of the previous posts, they will be presented to the user first. This happens because new posts are prioritized over posts the user has already seen. All "old" posts, meaning posts the user has already seen, maintain their position, unless there are new comments.

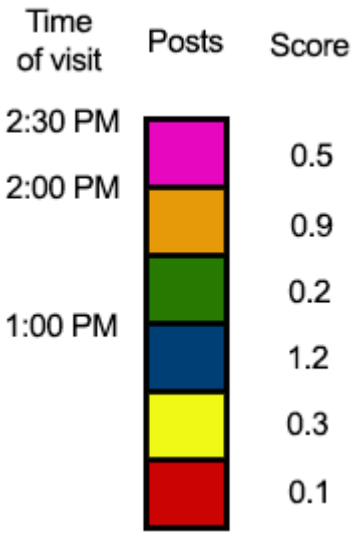

Figure 5.2: Example of ranking new posts [Bac16]

Users' engagement

The calculated score of a post will take into consideration its number of likes, comments and shares. Normally, people will share content that they think others should see, be it because it is useful or important. Comments suggest some level of discussion or acknowledgment, while likes are the form of feedback on the quality of a post that requires the least effort on behalf of the user.

In the proposed post ranking algorithm, it is considered that the degree of effort that users put into their interaction with certain posts in the timeline is a good indicator of how relevant the posts are. As such, the highest weight is placed in post shares (as they require selecting a post, sharing it and typing a message), followed by comments (requires typing a message) and lastly the likes (which requires only a single mouse click).

Prior interaction

Besides the users' engagement, the algorithm will also analyze the user's prior interactions with the projects. The more interactions (likes, comments, shares) the user has had with the posts of a project, including posts created by the user himself, the higher the score will be for a post from that project. The number of interactions with a given project will give us the user's level of involvement with that project. For example, if a user has commented one post of a project and liked another one, the number of interactions with that project will be 2. Users with the same posts may have a timeline ordered in a different way depending on this feature.

Post age

The "age" of a post is given by difference between the amount of time since the post was created and the time at the start of the post ranking operation. Newer posts will benefit from a higher "time score" in order to guarantee that older posts that are no longer relevant to the current activities of the project are not shown on top of newer posts, even if they have lots of activity (likes, comments, shares).

5.3 Ranking method

The implemented ranking method combines 5 different features: number of shares, comments and likes of the post being ranked, interactions performed by the current user with posts generated by the same project as the post being ranked, and the time score, a previously computed value that depends on the age of a post.

5.3.1 Time score

The time score is given by computing the multiplicative inverse of the difference between the current time c_t and the time of creation of the post t_c,

$$\frac{1}{c_t - t_c}$$

Figure 5.3 shows the time score varies with the time difference in a rectangular hyperbole. This way, recent posts will have a considerably higher time score when compared to older posts.

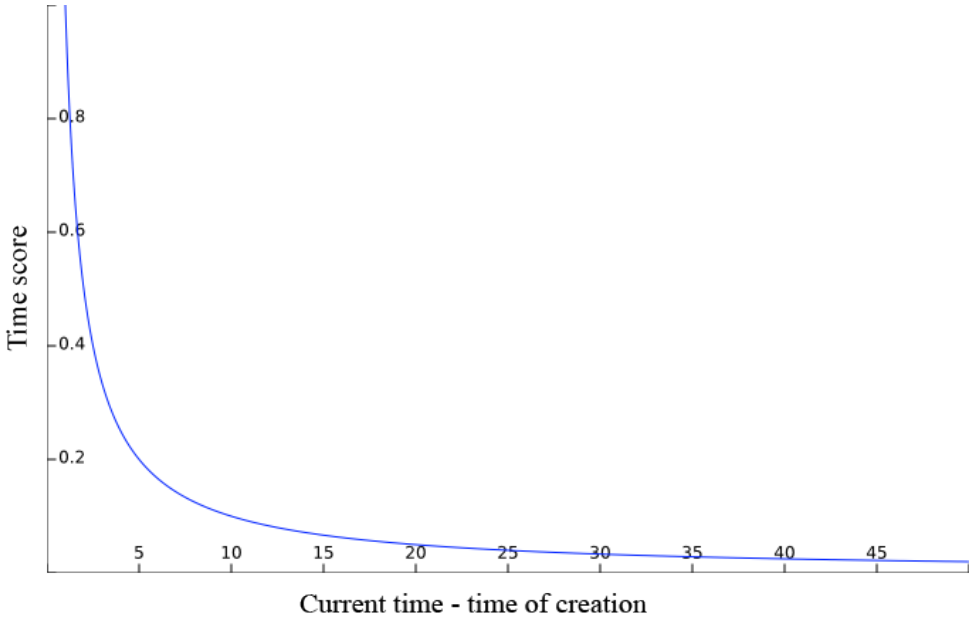

Figure 5.3: Reciprocal function

5.3.2 Normalization

The different post features selected for the ranking have different ranges of possible values, and thus need to be normalized. The different ranges of the features are:

- Likes: [0, number of contributors in project]
- Comments: [0, ∞]
- Shares: [0, ∞]
- Time score: [0, 1]

As we are seeking to combine these partial scores, we need to normalize them in order to have the same range of values for each of the inputs. For a given post, each feature is normalized by dividing the value of the feature by the maximum value of that feature in the set of posts that is being scored.

5.3.3 Post scoring formula

Taking into consideration what was previously described, we developed a formula that calculates the score for each post in a timeline. For a given post, and given the normalized number of shares s, comments c, likes l, project interactions p_i and the time score t_s, the final score of a post p is calculated as follows:

$$s(p) = 0.35s + 0.25c + 0.1l + 0.2p_i + 0.1t_s$$

The constants represent the weight given to each of the features in the final score. Shares will have a weight of 35%, followed by the comments with 25% and then project interactions with 20%. Both the likes and the time score have a weight of 10%.

5.4 Application of the method

In order to show how the ranking method works in an actual timeline, we simulated an interaction between two users on a project which generated 10 posts from different types. We also simulated some activity in the timeline, such as commenting and liking some posts. Figure 5.4 shows the resulting timeline chronologically ordered, with no ranking method applied. Figure 5.5 shows the resulting timeline with the ranking method applied. When the ranking method is applied, posts with more activity are shown at the top. Table 5.1 shows which posts are sent up, down or maintain the original chronological order when ranked. Table 5.2 presents the normalized value of each feature for each post and the final score of that post.

Table 5.1: Order comparison in the two timelines

Position	Ranked position	Difference	Post type	Likes	Comments	Shares
1	5	-4	metadata change	0	0	0
2	6	-4	metadata change	0	0	0
3	1	+2	manual	1	1	0
4	7	-3	create directory	0	0	0
5	8	-3	metadata change	0	0	0
6	2	+4	file upload	0	1	0
7	3	+4	metadata change	1	0	0
8	4	+4	file upload	1	0	0
9	9	0	file delete	0	0	0
10	10	0	file upload	0	0	0

Table 5.2: Posts' score in ranked timeline

Position	Likes	Comments	Shares	Project interactions	Time score	Score
1	1	1	0	1	0.124	0.562
2	0	1	0	1	0.116	0.462
3	1	0	0	1	0.113	0.311
4	1	0	0	1	0.110	0.311
5	0	0	0	1	1.000	0.300
6	0	0	0	1	0.387	0.239
7	0	0	0	1	0.121	0.212
8	0	0	0	1	0.118	0.212
9	0	0	0	1	0.110	0.211
10	0	0	0	1	0.109	0.211

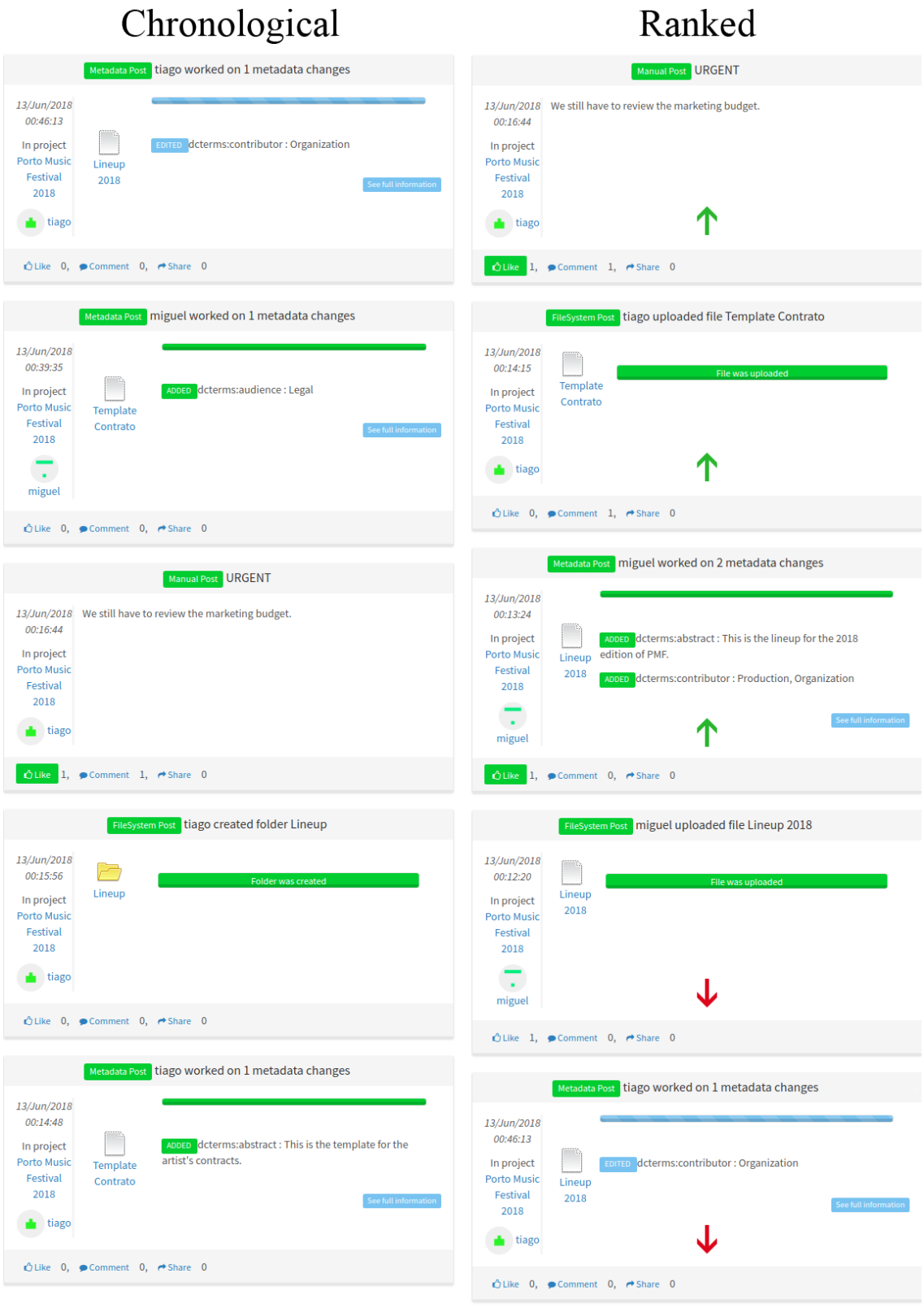

Figure 5.4: Comparison between the two timelines - page 1

(continued)

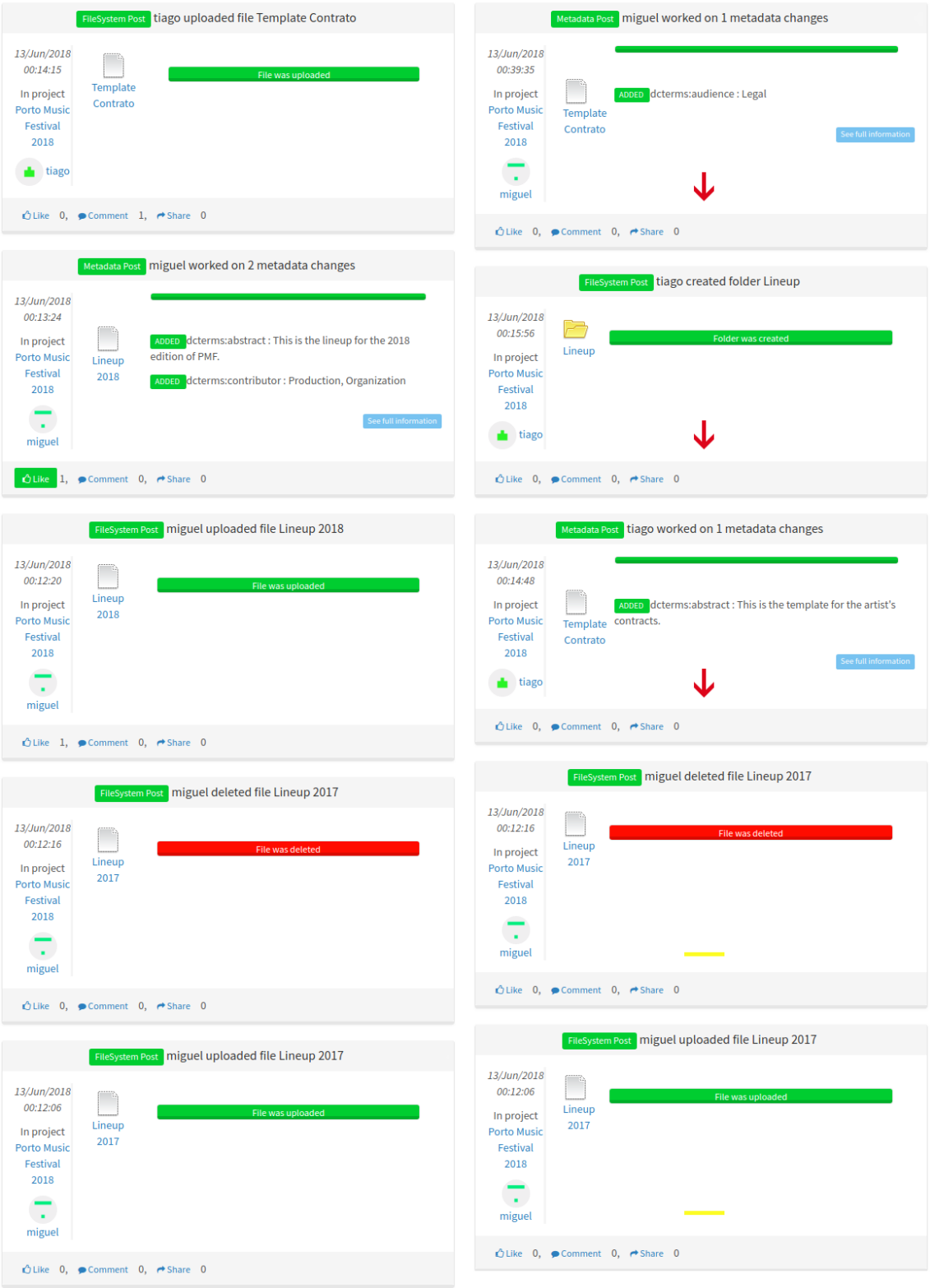

Figure 5.5: Comparison between the two timelines - page 2

Chapter 6

Implementation

This chapters explains how the solution was implemented, giving a high-level overview of the implementation. First, describes the used technologies (Sec. 6.1), then the implemented data model (Sec. 6.2) and how it was translated into a relational model (Sec. 6.3) and, finally, describes the interaction of a researcher with the system components (Sec. 6.4).

6.1 Technologies

Prior to this work, the Dendro platform already had a solid base and a well defined technology stack. The solution was implemented with the existing technologies, adding an Object-Relational Mapping (ORM) library.

The following technologies were used:

- **Node.js**[1] with **Express.js**[2], for the back-end;
- **AngularJS**[3], for the front-end;
- **MySQL**[4], to implement the necessary data model;
- **Sequelize**[5], a promise-based ORM.

Even though the base Dendro data model is implemented in SPARQL, with a Virtuoso Universal Server[6] working as server for the triplestore database, there was already a functionality using a MySQL database. Before implementing the solution, we first needed to change the existing code to use the Sequelize library to ensure the maintainability of the code in the future. Moreover, there

[1] https://nodejs.org/en/
[2] https://expressjs.com/
[3] https://angularjs.org/
[4] https://www.mysql.com/
[5] http://docs.sequelizejs.com/
[6] https://virtuoso.openlinksw.com/

was a single query in the entire system to insert new data in a single table. Given the modifications made to the database to support the ranking module, an ORM solution had to be introduced. Sequelize added, among other important solutions, object mappers, making data access more abstract and portable, and database migrations.

6.2 Data model

The implemented post ranking algorithm uses data from multiple sources such as posts, users and projects. In order to improve query performance, we decided to design a relational model, providing a single common data model for all the necessary data. This data model saves only the identifiers for the resources as that is all that is required in the implemented queries. Not only that, but doing otherwise would lead to redundancy, as the information of the posts is already in the graph and keeping the information updated in both the graph and the MySQL server would generate a higher and unnecessary server load. Figure 6.1 presents the unified modeling language (UML) diagram for the data model.

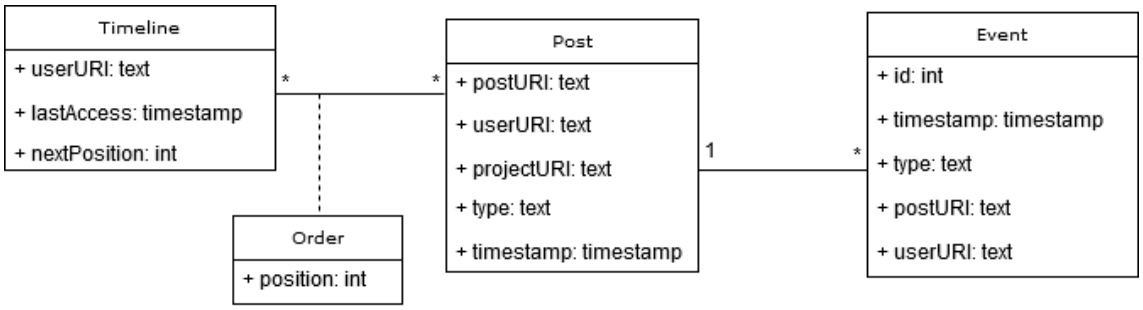

Figure 6.1: Data model

In order to relate the registries in the Dendro graph and the MySQL database, we use the URIs, which are the unique identifiers for each resource.

For each post, we save its URI, the URI of the user that created the post, the URI of the project it was generated from, the type (manual, metadata change, file upload, file deletion, creation of directory, deletion of directory) and the timestamp.

For each event, we save the URI of the user that generated the event, the URI of the post where the event was generated, the type (like, comment, share, creation of a post) and the timestamp.

For each timeline, we save the URI of the user (each user has only one timeline), a timestamp that indicates when the user last accessed the timeline and the "next position", meaning the position of the next post to be inserted in the timeline.

Everytime there is a new like, comment, share or post, either manual or generated by the system, a new event is created and inserted in the database. In the case of a post, it is also inserted in the table of posts. Posts are associated to the timeline of a specific user by the position of the post in that timeline.

6.3 Relational model

Figure 6.2 presents the relational model for the data model described in Section 6.2.

Table "timeline_posts" was created in order to map the many-to-many relation between a timeline and the posts in that timeline. This table saves the identifier for the timeline and the URI of the post, as well as the position of the post in the timeline.

Tables "post_types" and "event_types" were created in order to reduce data redundancy and improve data integrity and map a one-to-many relation with tables "posts" and "events", respectively.

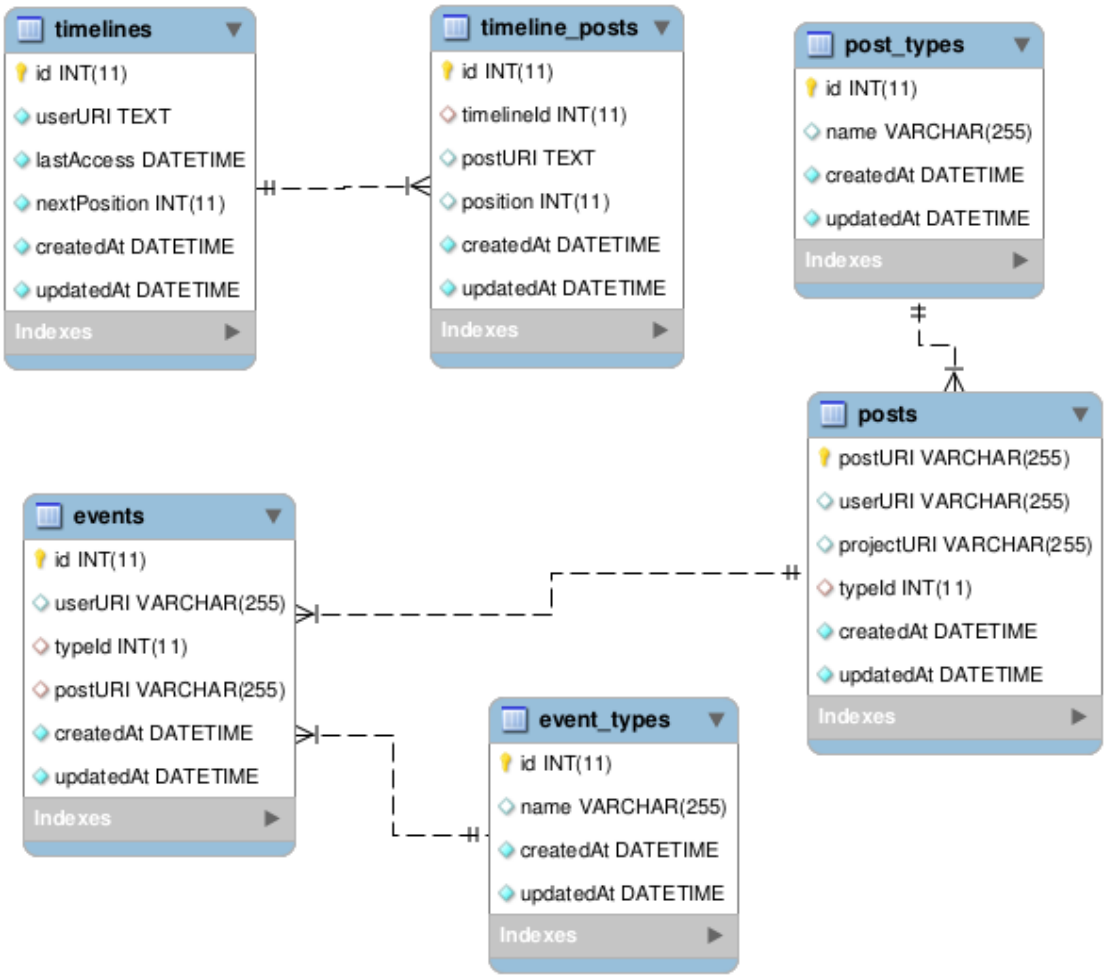

Figure 6.2: Relational model

6.4 Interaction between system components

The interaction between the researcher and the timeline in Social Dendro is shown in Figure 6.3. Depending on the route used to access the timeline, the controller "timelineCtrl" initializes the

timeline with a flag "useRank" that is true if the researcher accessed the ranked timeline and false otherwise. This flag is passed in subsequent calls, first to the timeline service and then in the HTTP request made to the back-end.

The back-end processes the request and calls the respective handler in controller "posts". If the user wants the ranked posts, the handler will call the function that scores and ranks the posts. If not, the handler will call the function that returns the posts URIs in chronological order. After determining the URIs of the posts to be shown to the researcher and the respective order, the controller will call the function that retrieves the information from each post from the Dendro graph, as it is not replicated in the relational database used by the ranking algorithm.

The information from the posts is then sent to the researcher and rendered in the timeline view.

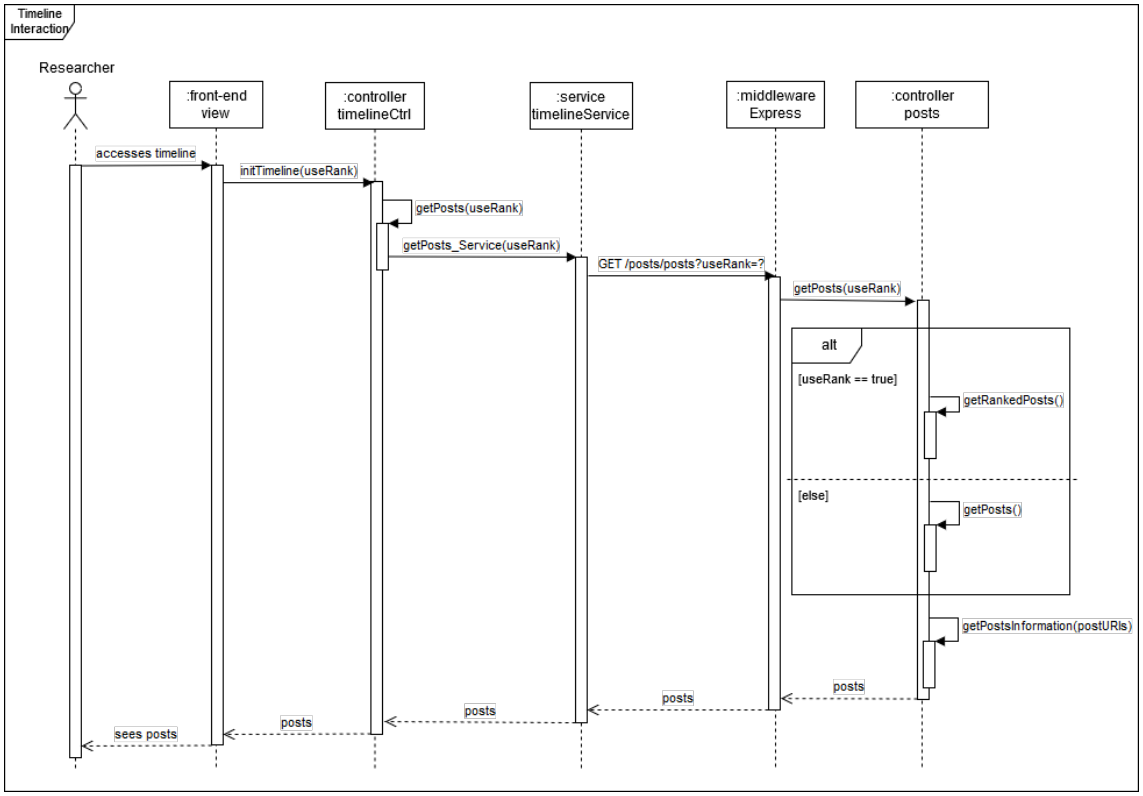

Figure 6.3: Sequence diagram representing an interaction with the timeline

Chapter 7

Evaluation

This chapter describes the evaluation methodology. It starts with a brief introduction (Sec. 7.1), then describes the required changes to the interface (Sec. 7.2) and the profile of the participants in the experiment (Sec. 7.3) and proceeds to describe and analyze the results of the initial questionnaire (Sec. 7.4) and the tasks the users were asked to do (Sec. 7.5). Then, we explain the evaluation metric (Sec. 7.6) and, finally, we analyze and interpret the results of the evaluation (Sec. 7.7) and present the results of the final questionnaire (Sec. 7.8).

7.1 Introduction

A successful implementation of the solution will be directly related to the quality of the interaction users will have with it. Considering this, we need to devise a test capable of properly evaluating the quality of the user's interaction with the timeline.

Our experimental scenario requires the simultaneous interaction of two users on the same project, with one evaluator making sure the users are executing the set of tasks properly.

In this chapter, we describe the experiment made, as well as the obtained results and possible changes to the implemented solution.

Users will have to answer 2 questionnaires, one before and one after the experiment. For the experiment, users will follow a script that describes the tasks step by step. The questionnaires can be found in Appendix B and the script in Appendix A.

7.2 Changes to the interface

For this evaluation, some changes to the existing interface were required. We added a button to change between timelines and added the timeline title at the top, as shown in Figure 7.1. The check mark next to the timeline title indicates which timeline is selected.

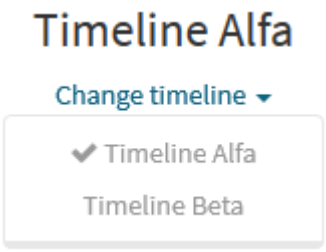

Figure 7.1: Timeline button

We also had to add two buttons to the posts' interface that allow the user to change the order of the respective post. These two buttons are marked with a red a box in Figure 7.2. The arrow up button makes the post go up one position, while the arrow down button makes the post go down one position.

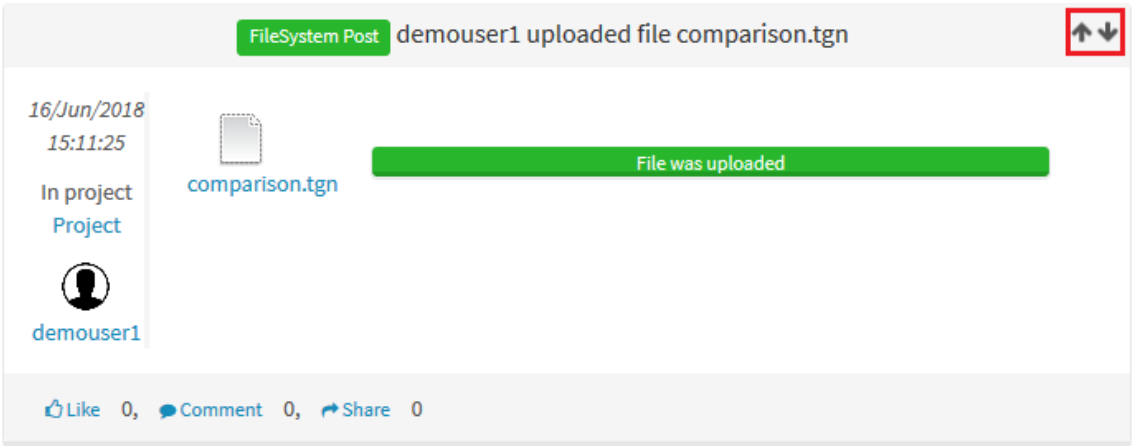

Figure 7.2: Post's button

7.3 Profile of the participants

In order to test the implemented solution, we asked 14 users to use the platform and evaluated their performance. The basic information of the users is presented in table 7.1.

Table 7.1: User's information

ID	Age	Gender	Ocupation
U1	23	Male	Student
U2	23	Female	Student
U3	20	Male	Student
U4	19	Male	Student
U5	21	Male	Student
U6	31	Male	Professor
U7	26	Male	Researcher
U8	14	Male	Student
U9	29	Female	Worker
U10	22	Male	Student
U11	23	Female	Student
U12	22	Male	Student
U13	22	Male	Student
U14	22	Male	Student

7.4 Initial questionnaire

In order to properly identify and characterize the evaluated users, they had to answer a questionnaire (Appendix B) before executing the tasks. In the questions where users had to answer with a number from 1 to 5, 1 meant the lowest level and 5 the highest.

Table 7.2: Questions from the initial questionnaire

ID	**Description**
QI1	Classify how frequently you use the Dendro platform (1-5)
QI2	Classify how frequently you use social networks (1-5)
QI3	Indicate your level of experience in the subject of research data management (1-5)

With question QI1 we intended to find out the level of experience of the participants in the Dendro platform. Most of the sample (85.7%) had no previous experience with the Dendro platform, while the remaining 14.3% used the platform regularly.

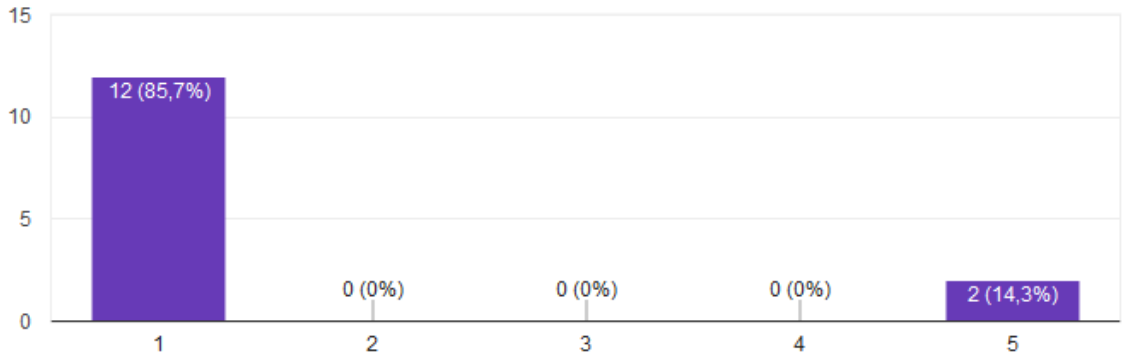

Figure 7.3: Results from question QI1

Answers to question QI2 revealed most of the participants used social networks on a regular basis (85.7%), with only one of the participants not using them at all.

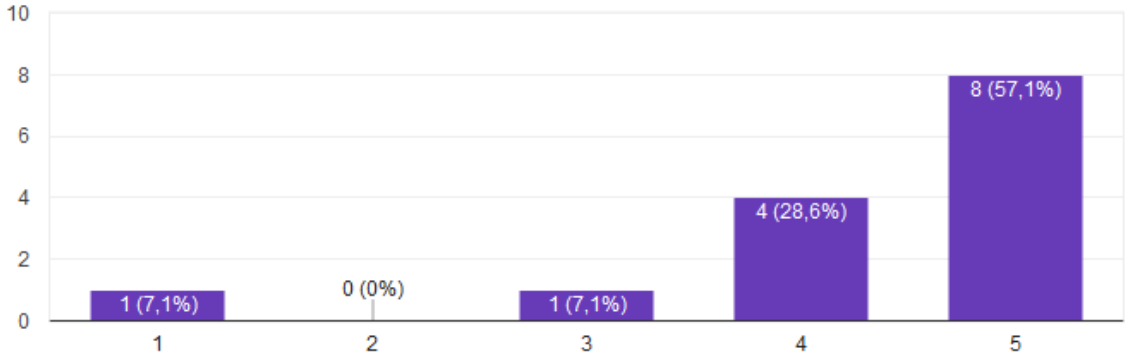

Figure 7.4: Results from question QI2

In question QI3, most of the participants (71.4%) indicated they had no to little experience in research data management, while one indicated he was proficient. 21.4% had a medium level of experience in research data management.

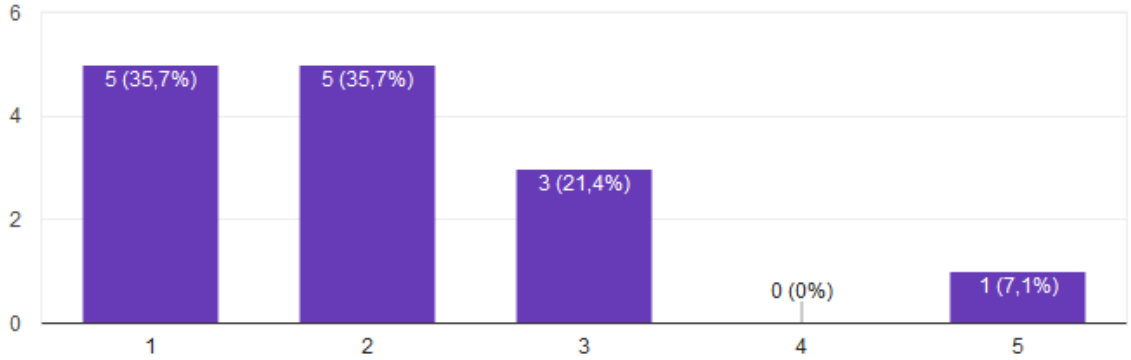

Figure 7.5: Results from question QI3

7.5 Tasks

The objective of the tasks was to generate the maximum possible number of different posts and interactions, while not making the experiment too long, both in terms of duration and number of posts to analyze.

Both users are required to create a new user in the beginning of the experiment in order to ensure they both have an empty timeline. Users were also left the choice to decide who would be User A and User B, as this does not affect the performance of the users or the outcome of the experiment.

Some of the tasks for User A required tasks from User B to be completed first and vice-versa. In these cases, the evaluator made sure the users would wait for each other before proceeding to the next task. This guarantees equal circumstances for all the experiments, as this is the only way to have a consistent evaluation.

Even though it would be easier and less time consuming to present the users with dummy posts and ask them to classify them, we opted not to, because users do not have any kind of relation to them, as the posts would be about operations the users did not make and files they do not know. It would not make sense to ask users to reorder posts about events they did not see happen and thus do not know why they appear in the timeline.

Table 7.3: Tasks for User A

ID	Description
TA1	Create a new user in Dendro
TA2	Create a new project named "Porto Music Festival 2018" and add User B as a collaborator
TA3	Upload file "Lineup 2017" and then delete the file
TA4	Upload file "Lineup 2018" and add the descriptor "abstract", briefly describing the file, and the descriptor "contributor", naming the entities responsible for contributing to the file
TA5	Access the timeline, comment the file uploaded by User B and like a post
TA6	Analyze timeline Alfa
TA7	Reorder the posts from timeline Alfa as you wish, using the arrows on the top right corner
TA8	Analyze timeline Beta
TA9	Reorder the posts from timeline Beta as you wish, using the arrows on the top right corner (do not worry about making sure the order is the same in both cases)
TA10	Delete descriptor "contributor" from file "Lineup 2018"
TA11	Add a new descriptor to file "Lineup 2018"
TA12	Like a post
TA13	Analyze timeline Beta
TA14	Reorder the posts from timeline Beta as you wish, using the arrows on the top right corner
TA15	Analyze timeline Alfa
TA16	Reorder the posts from timeline Alfa as you wish, using the arrows on the top right corner (do not worry about making sure the order is the same in both cases)

Table 7.4: Tasks for User B

ID	Description
TB1	Create a new user in Dendro
TB2	Verify you were added as contributor to project "Porto Music Festival 2018"
TB3	Upload file "Contract Template" and then add the descriptor "abstract", briefly describing the file
TB4	Create folder "Lineup" and move file "Lineup 2018", uploaded by User A, to the folder
TB5	Access the timeline, make a manual post with some text related to the project and like a post
TB6	Analyze timeline Beta
TB7	Reorder the posts from timeline Beta as you wish, using the arrows on the top right corner
TB8	Analyze timeline Alfa
TB9	Reorder the posts from timeline Alfa as you wish, using the arrows on the top right corner (do not worry about making sure the order is the same in both cases)
TB10	Edit descriptor "abstract" from file "Contract Template"
TB11	Share a post with some text
TB12	Comment a post
TB13	Analyze timeline Alfa
TB14	Reorder the posts from timeline Alfa as you wish, using the arrows on the top right corner
TB15	Analyze timeline Beta
TB16	Reorder the posts from timeline Beta as you wish, using the arrows on the top right corner (do not worry about making sure the order is the same in both cases)

Table 7.3 and Table 7.4 describe the tasks User A and User B were asked to perform, respectively. Timeline Alfa refers to the chronologically ordered timeline, while timeline Beta refers to the timeline with the ranking method applied. There are two moments where users are asked to reorder the timeline. In the first moment, we asked User A to reorder timeline Alfa first and then timeline Beta, while User B had to reorder timeline Beta first and then timeline Alfa. In the second moment, we asked user A to first reorder timeline Beta and then timeline Alfa, while User B reordered timeline Alfa first and then timeline Beta. This was done in order to remove any bias in the reordering that might come from seeing one of the timelines first.

7.6 Evaluation metric

The two timelines will be compared by averaging the difference between the position given by the system and the position given by the user. Table 7.5 takes the example presented in Figure 7.6 and shows the calculated difference d between the position given by the system and the position given by the user for each post. The average difference can then be calculated by the average of the sum of the absolute value of the differences,

$$\overline{d} = \frac{\sum_{i=1}^{4} d_i}{4} = \frac{3+0+2+1}{4} = 1.5$$

This means that, on average, users corrected the timeline presented by the system in 1.5 positions.

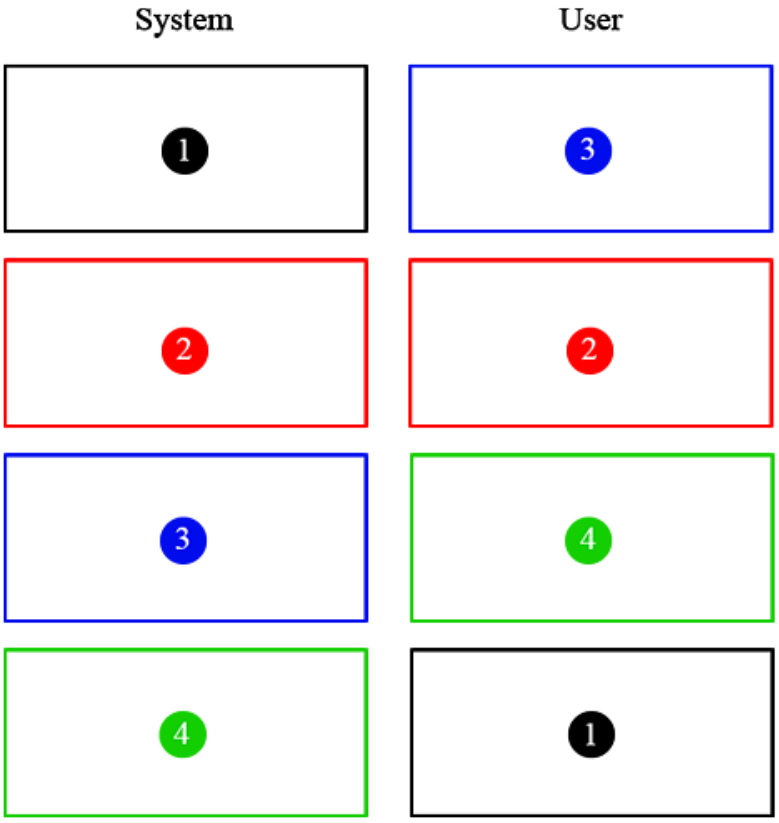

Figure 7.6: Example of a reordering

Table 7.5: Position difference between system and user

ID	System position	User position	Difference
1	1	4	3
2	2	2	0
3	3	1	2
4	4	3	1

7.7 Results

In order to evaluate the impact of the ranked timeline, we took into consideration the difference between the position attributed by the participant and the original position for each post in both

timelines. Table 7.6 shows the average difference between positions for both timelines, as well as the sum of the differences.

Table 7.6: Average position difference

	Mean	**Total**
Ranked	1.05	149
Unranked	1.41	200
Improvement	0.25	51

In this experiment, the ranked timeline showed a 25% improvement in the number of differences. This means the order of the ranked timeline is 25% more accurate in terms of the users' preferences.

During the experiment, some users commented that some post types might not be as interesting as others. Table 7.7 shows the average differences by post type and the respective improvement in the ranked timeline. The lack of improvement in the "file delete" post type can be explained by the fact that some users chose to comment that post, but still preferred it to be shown at the bottom of the timeline. In the ranked timeline, as the post had a comment, it was shown in the first positions of the timeline.

The lack of improvement in the "manual" post type can be explained by the fact the users were making manual posts that were not related to the project's theme, so in the reordering process that post was regarded as not important.

Some users also pointed that posts with no comments that refer to changes made by the own user may not be as interesting as the posts from others users, because the user is aware of the changes made by its own actions and thus does not need to see those posts first.

Table 7.7: Average position difference by post type

	Mean		
	Ranked	**Unranked**	**Improvement**
File upload	0.64	1.17	0.45
File delete	1	0.92	-0.09
Metadata change	1.15	1.42	0.19
Make directory	1.08	1.91	0.43
Manual	2.17	2	-0.08
Share	0.5	1.5	0.67

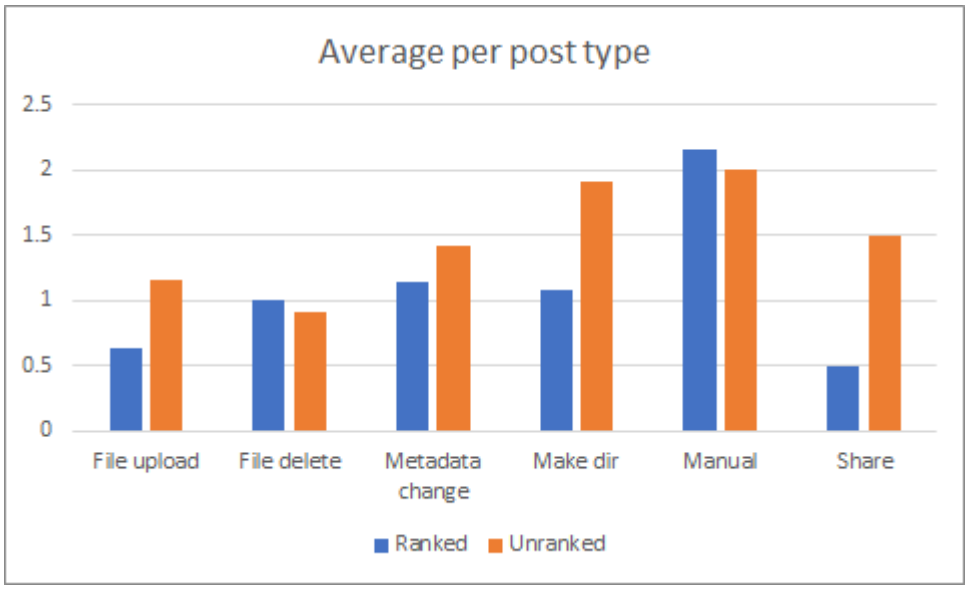

Figure 7.7: Chart representing average position difference by post type

7.8 Final questionnaire

In order to get some feedback from the participating users about the experiment and what they valued more, users had to answer a questionnaire (Appendix B) at the end of the experiment. In the questions where users had to answer with a number from 1 to 5, 1 meant the lowest level and 5 the highest. For questions 1 and 2, the lowest level was "not important" and the highest level was "really important". For questions 3, 4 and 5 the lowest level meant "strongly disagree" while the highest meant "strongly agree".

Question QF1 reveals participants have the opinion that comments highlight the importance of a certain post the most, with all participants but one choosing either 4 or 5. Share is the following component with more 4 and 5, but with 2 participants choosing 2. Participants also attributed some importance to the date of creation of the post, with 9 participants choosing either 4 or 5, and the remaining 5 choosing 3. Like is the less unanimous component with 2 participants choosing 2, 5 participants choosing 5, 4 participants choosing 4 and 3 participants choosing 3.

Table 7.8: Questions from the final questionnaire

ID	Description
QF1	Classify the following elements (like, comment, share, creation date) in terms of their importance for the valorization of a post (1-5)
QF2	Classify the importance/influence of each type of post (manual, metadata modification, file upload, file delete, create directory, delete directory) to the perception of the activities of a project (1-5)
QF3.1	Timeline Alfa shows me interesting information (1-5)
QF3.2	Timeline Alfa shows me interesting information in the pretended order (1-5)
QF3.3	Timeline Alfa allows me to rapidly perceive the activity of the project (1-5)
QF3.4	Timeline Alfa allows for a greater interaction between the project collaborators (1-5)
QF3.5	Timeline Alfa has posts that could be omitted (1-5)
QF4.1	Timeline Beta shows me interesting information (1-5)
QF4.2	Timeline Beta shows me interesting information in the pretended order (1-5)
QF4.3	Timeline Beta allows me to rapidly perceive the activity of the project (1-5)
QF4.4	Timeline Beta allows for a greater interaction between the project collaborators (1-5)
QF4.5	Timeline Beta has posts that could be omitted (1-5)
QF5	In a timeline, presenting the most interesting posts first leads to a higher interaction with it (1-5)
QF6	In what way can the the user experience with the timeline be improved? In case you have a suggestion, leave it below

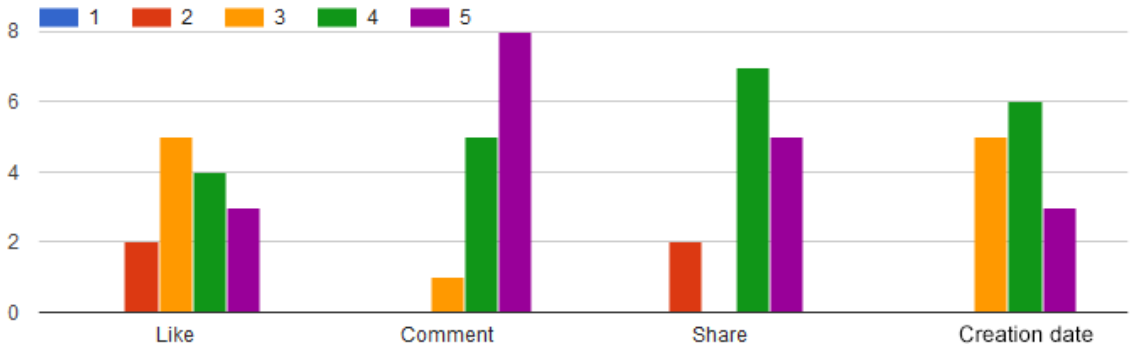

Figure 7.8: Results from question QF1

With question QF2 we can understand which types of posts the participants find more important to understand the activities of a project. It becomes clear that the deletion of files and directories and the creation of directories is not as important to the participants as the others types of posts, with most of the participants giving an importance of 2 to these types of posts. Participants attributed similar importance to file uploads and metadata changes, with manual posts being the most important with 11 participants choosing either 4 or 5.

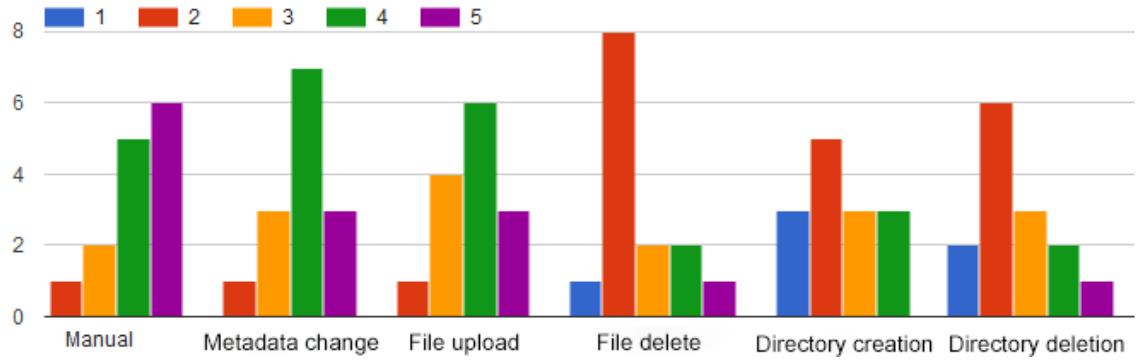

Figure 7.9: Results from question QF2

Questions QF3 and QF4 looked to compare both timelines. The results here were not conclusive as most of the participants pointed they didn't notice much of a difference between the two timelines. This can be due to the fact that the users had no prior knowledge they were going to have to compare the two of them and thus were not paying much attention to the differences between them. Also, differences between the two timelines would perhaps be more noticeable with continued use and with projects with a lot of activity..

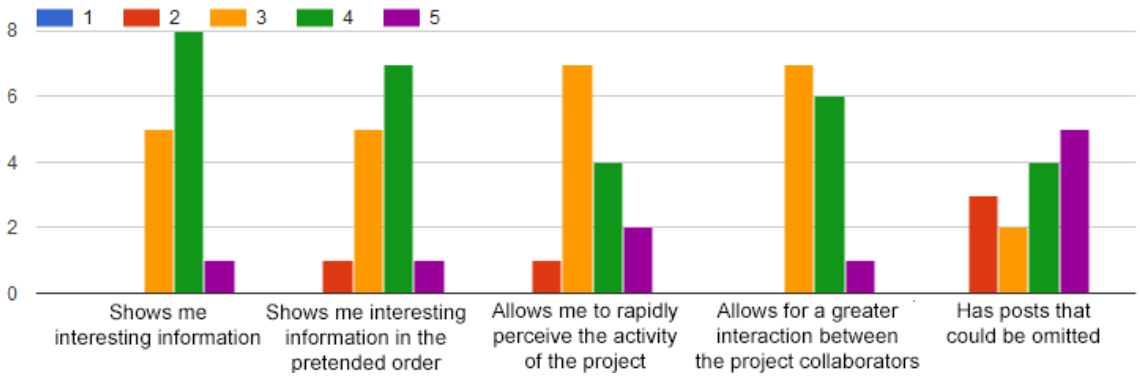

Figure 7.10: Results from question QF3

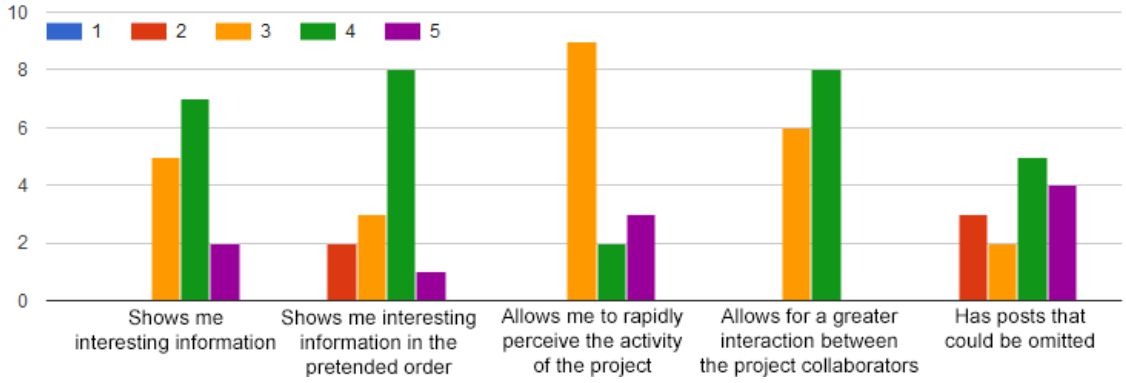

Figure 7.11: Results from question QF4

The results from question QF5 show that most participants (11) agree with the idea that presenting the most interesting posts first leads to a higher interaction with the timeline.

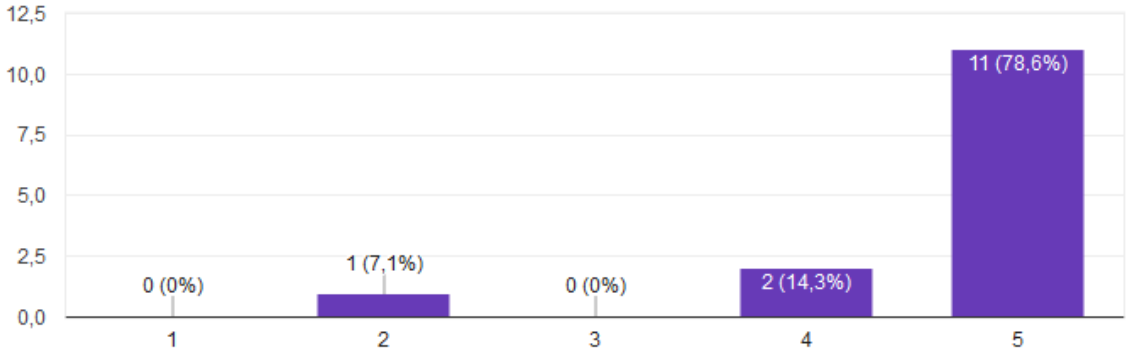

Figure 7.12: Results from question QF5

Answers to the question QF6 are presented in Table 7.9.

Table 7.9: Results from question QF6

ID	Description
A1	"There could be buttons to send a post directly to the top/bottom instead of one by one"
A2	"The most recent activities should be at the top"
A3	"The most recent posts and the most interesting content should be at the top"
A4	"Posts created by my work in Dendro (or even my manual posts) should be presented more to the bottom of the timeline. I'm more interested in knowing what the other contributors to the project have done. However, if my posts have comments or shares I would like to see them first so that it is easier and faster to reply to the comments. Maybe show first posts that represent deletion of content (either files/-folders/metadata) versus posts representing new folders/files/descriptors, because in my opinion this is useful to correct the action of deleting something that wasn't supposed to be deleted"
A5	"Hide some types of posts. Make posts' layout more compact so that there's no unused space. Displaying people's real names and the labels for the descriptors instead of "npereira" or "dc:title". Display a reference to the project to which the file or folder belongs to. Place in the bottom of the timeline some types of posts such as creation of folder"
A6	"Allowing to save or mark posts that are interesting to my project so I can see them faster"
A7	"Posts could have a different visual representations according to their type"

Some of the suggestions given by participants were related to changes in the interface, which is not directly related to the implemented solution.

One of the suggestions to place the deletion of content at the top goes against the opinion of most of the participants, as shown with the results of question QF2.

Chapter 8

Conclusions and Future Work

This chapter presents the final balance (Sec. 8.1), followed by the possible future work (Sec. 8.2).

8.1 Final balance

The generation of quality research data is directly connected to the way that data is managed along the research process. In that sense, the adoption of good research data management practices is a key factor in the generation of properly described data that can be used by other investigators without previous knowledge of the context of that data.

The Dendro platform aims to involve researchers in the description and management of their data in the early stages of research by implementing an easy-to-use interface and a triple store-based data model that allows researchers, typically without data management skills, to expand it by loading ontologies that specify domain-specific metadata descriptors. To further promote collaboration and data description, Social Dendro implements some social-network concepts, such as likes, comments and shares as a way to give feedback on the developments of the investigation in a fast and practical manner.

This work builds on the Dendro platform, and more specifically on Social Dendro, by conceiving and implementing an algorithm capable of properly ranking posts in terms of relevance and presenting a timeline of posts ordered such that the user is able to see and interact first with what he or she finds more relevant. In order to evaluate the quality of the solution, we also performed an experiment with a number of users.

The objectives of this dissertation were achieved, as the ranking algorithm was properly implemented and the performed experiment showed that the ranked timeline is closer to the user's preferred order.

8.2 Future work

After the evaluation, we concluded that the algorithm could continue to be refined, by adding new features in the formula such as a score dependent on the post type and taking into consideration that posts from the own user may not be as interesting because the user is aware of the change made by its own actions. We could also introduce a filtering feature, hiding some unnecessary posts.

The timeline interface can also be improved so that the user can better distinguish between post types and properly understand the activities of a project.

8.2.1 Evaluation

The experiment required the interaction of 2 users and had a certain degree of complexity, especially for users with no experience with the Dendro platform, which caused some confusion with what was supposed to be done and what was being evaluated. On top of that, the experiment did not allow us to evaluate a continued use situation, where the implement algorithm would perhaps be more useful. Some more variables could be evaluated in order to draw other conclusions, for example if there was a relation between the post's author and the position given by the user. Research work needs to be done in order to better this evaluation.

It is common to see ranking experiments being performed using a ground truth ranking. In this case, there is no pre-established ground-truth, because different experiment runs can generate different sets of posts to be ranked, i.e. there is no shared test collection across experiment sessions. We only wanted a single contact with each of the users to not overburden them.

In the future, an experiment with a ground-truth could be designed if it was split across two sessions for each user, where they would first rank posts as outlined in this experiment, and then rank the posts present in the other users' timelines. After obtaining a sufficiently high number of manually ranked timelines, we could determine an average position for each post, for example, and therefore establish a "ground-truth" timeline to compare our rankings against.

A possible solution would be to ask our users to rank a set or randomly sorted posts in addition to the two timelines they saw (chronologically-ranked and algorithm-ranked) in order to build a "ground-truth" over which to compare the rankings in each timeline. We did not do this because (1) it would add another burden to an already time-consuming experiment and (2) this "ground-truth" would only be valid for the two users in each experiment, as different experiments generate different sets of posts, as said before.

By avoiding this manual ranking of randomly ordered posts and directly comparing the distance between rankings in the two cases we ensured that the same posts were present to be ranked.

References

[ACC+15] Massimiliano Assante, Leonardo Candela, Donatella Castelli, Paolo Manghi, and Pasquale Pagano. Science 2.0 Repositories: Time for a Change in Scholarly Communication. *D-Lib Magazine*, 21(1/2):1, jan 2015.

[ATD17] Omar Alonso, Serge-Eric Tremblay, and Fernando Diaz. Automatic Generation of Event Timelines from Social Data. In *Proceedings of the 2017 ACM on Web Science Conference - WebSci '17*, pages 207–211, New York, New York, USA, 2017. ACM Press.

[Bac16] Lars Backstrom. Serving a Billion Personalized News Feeds. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining - WSDM '16*, pages 469–469, New York, New York, USA, 2016. ACM Press.

[BHBL09] Christian Bizer, T Heath, and T Berners-Lee. Linked data-the story so far. *International journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[BHIBL08] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (LDOW2008). In *Proceeding of the 17th international conference on World Wide Web - WWW '08*, page 1265, New York, New York, USA, 2008. ACM Press.

[Biz12] Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from\nWikipedia. 1:1–5, 2012.

[BL06] Tim Berners-Lee. Linked Data. 2006.

[Bor12] Christine L. Borgman. The conundrum of sharing research data. *Journal of the Association for Information Science and Technology*, 63(6):1059–1078, 2012.

[BP98] S Brin and L Page. The anatomy of a large scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1/7):107–17, 1998.

[CCAB12] Giovanni Comarela, Mark Crovella, Virgilio Almeida, and Fabricio Benevenuto. Understanding factors that affect response rates in twitter. In *Proceedings of the 23rd ACM conference on Hypertext and social media - HT '12*, page 123, New York, New York, USA, 2012. ACM Press.

[dSCRL14] JR da Silva, JA Castro, C Ribeiro, and JC Lopes. The Dendro Research Data Management Platform: Applying Ontologies to Long-Term Preservation in a Collaborative Environment. In Serena Coates, Ross King, Steve Knight, Christopher A. Lee 0001, Peter Mckinney, Erin O'meara, and David Pearson, editors, *Proceedings of the 11Th International Conference on Digital Preservation, Ipres 2014, Melbourne, Australia, October 6 - 10, 2014*, Ipres, 2014. Citations: dblp.

REFERENCES

[dSRL14] João Rocha da Silva, Cristina Ribeiro, and João Correia Lopes. Ontology-based multi-domain metadata for research data management using triple stores. In *Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14*, pages 105–114, New York, New York, USA, 2014. ACM Press.

[FFYZ16] Feifan Fan, Yansong Feng, Lili Yao, and Dongyan Zhao. Adaptive Evolutionary Filtering in Real-Time Twitter Stream. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16*, pages 1079–1088, New York, New York, USA, 2016. ACM Press.

[GB14] Ramanathan Guha and Dan Brickley. RDF schema 1.1. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[Hig08] Sarah Higgins. The DCC Curation Lifecycle Model. *International Journal of Digital Curation*, 3(1):134–140, 2008.

[JAK12] Lori Jahnke, Andrew Asher, and Spencer D C Keralis. *The Problem of Data*. Number August. 2012.

[KCdS+17] Yulia Karimova, João Aguiar Castro, João Rocha da Silva, Nelson Pereira, and Cristina Ribeiro. Promoting Semantic Annotation of Research Data by Their Creators: A Use Case with B2NOTE at the End of the RDM Workflow. In *Communications in Computer and Information Science*, volume 755, pages 112–122. 2017.

[MT04] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. *Proceedings of EMNLP*, 85:404–411, 2004.

[P. 08] P. Bryan Heidorn. Shedding Light on the Dark Data in the Long Tail of Science. *Library Trends*, 57(2):280–299, 2008.

[PDF07] Heather A. Piwowar, Roger S. Day, and Douglas B. Fridsma. Sharing Detailed Research Data Is Associated with Increased Citation Rate. *PLoS ONE*, 2(3):e308, mar 2007.

[PdSR17] Nelson Pereira, João Rocha da Silva, and Cristina Ribeiro. *Social Dendro: Social Network Techniques Applied to Research Data Description*, pages 566–571. Springer International Publishing, Cham, 2017.

[PSRP+12] Peter Patel-Schneider, Sebastian Rudolph, Bijan Parsia, Pascal Hitzler, and Markus Krötzsch. OWL 2 web ontology language primer (second edition). W3C recommendation, W3C, December 2012. http://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[RARC14] João Rocha da Silva, João Aguiar Castro, Cristina Ribeiro, and João Correia Lopes. Dendro: Collaborative Research Data Management Built on Linked Open Data. In *Proceedings of the 11th European Semantic Web Conference*, pages 483–487. 2014.

[SR15] Alisa Surkis and Kevin Read. Research data management. *Journal of Medical Library Association*, 10(3):154–156, 2015.

[UG96] Mike Uschold and Michael Gruninger. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11(02):93, jun 1996.

REFERENCES

[WJ11] Angus Whyte and Tedds Jonathan. Making the Case for Research Data Management. *A Digital Curation Centre Briefing Paper*, (September):1–8, 2011.

[WV14] Wesley Waldner and Julita Vassileva. Emphasize, don't filter! In *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*, pages 313–316, New York, New York, USA, 2014. ACM Press.

[YFFZ16] Lili Yao, Feifan Fan, Yansong Feng, and Dongyan Zhao. Leveraging Tweet Ranking in an Optimization Frameworkfor Tweet Timeline Generation. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries - JCDL '16*, number 2, pages 245–246, New York, New York, USA, 2016. ACM Press.

[YLLR12] Min-Chul Yang, Jung-Tae Lee, Seung-Wook Lee, and Hae-Chang Rim. Finding interesting posts in Twitter based on retweet graph analysis. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, number 2, page 1073, New York, New York, USA, 2012. ACM Press.

REFERENCES

Appendix A

Script

Guião

Descrição da plataforma

O Dendro é uma plataforma colaborativa open-source para a gestão de dados de investigação, atualmente a ser desenvolvida na FEUP. Os grupos de investigadores têm a possibilidade de criar projetos, onde depois podem armazenar e descrever os seus dados. O Social Dendro, a extensão social do Dendro, foi desenvolvido aplicando à plataforma as técnicas comuns de redes sociais. Consiste numa timeline de posts que representam as alterações aos projetos dos utilizadores, tais como adições/remoções de ficheiros e adições/edições de metadados, o que permite aos utilizadores ter uma visão clara das atividades do projeto. Os utilizadores podem ainda fazer posts manuais e dar "like", comentar e partilhar posts.

Preparação das tarefas

- A realização destas tarefas requer a interação simultânea de dois utilizadores;
- Os utilizadores terão de realizar tarefas diferentes, portanto é necessário definir previamente o Utilizador A e o Utilizador B;
- Nas tarefas em que é necessário fazer upload de um ficheiro, deve criar um ficheiro de texto com o nome pedido (o ficheiro não pode estar vazio), caso ele não exista;
- O objetivo é simular uma interação o mais realista possível;
- Ambos os utilizadores têm de responder a este questionário antes de iniciarem as tarefas: https://goo.gl/forms/kPXnRTvRsAJcpIxQ2.

Link para a plataforma: http://socialdendro.fe.up.pt

Tarefas

Utilizador A

1. Criar um novo utilizador no Dendro;
2. Criar um novo projeto com o nome "Porto Music Festival 2018" e adicionar o Utilizador B como colaborador;
3. Fazer upload do ficheiro "Lineup 2017" e de seguida apagar o ficheiro (lineup do ano errado);
4. Fazer upload do ficheiro "Lineup 2018" e adicionar um descritor "abstract" que descreva resumidamente o ficheiro e um descritor "contributor" que indique as entidades responsáveis por fazerem contribuições para o recurso (produção, organização);
   Esperar que o Utilizador B conclua a tarefa 4
5. Aceder à timeline, comentar o ficheiro carregado pelo Utilizador B (Template Contrato) e dar like num post;
   Esperar que o Utilizador B conclua a tarefa 5 e garantir que a timeline está atualizada
6. Analisar a timeline Alfa;
7. Reordenar os posts da timeline Alfa de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts;

8. Analisar a timeline Beta;
9. Reordenar os posts da timeline Beta de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts; (não se preocupe em assegurar a mesma ordem nos dois casos);
   Esperar que o Utilizador B conclua a tarefa 9
10. Apagar o descritor “contributor” do ficheiro “Lineup 2018”;
11. Adicionar mais um descritor ao ficheiro “Lineup 2018”;
    Esperar que o Utilizador B conclua a tarefa 11 e garantir que a timeline está atualizada
12. Dar like num post;
    Esperar que o Utilizador B conclua a tarefa 12 e garantir que a timeline está atualizada
13. Analisar a timeline Beta;
14. Reordenar os posts da timeline Beta de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts;
15. Analisar a timeline Alfa;
16. Reordenar os posts da timeline Alfa de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts; (não se preocupe em assegurar a mesma ordem nos dois casos);

Utilizador B

1. Criar um novo utilizador no Dendro;
   Esperar que o Utilizador A conclua a tarefa 2
2. Verificar que foi adicionado ao projeto como colaborador ao “Porto Music Festival”;
3. Fazer upload do ficheiro “Template Contrato” e adicionar um descritor “abstract” que descreva resumidamente o ficheiro;
   Esperar que o Utilizador A conclua a tarefa 4
4. Criar uma pasta “Lineup” e colocar lá dentro o ficheiro “Lineup 2018”, carregado pelo Utilizador A;
5. Aceder à timeline e fazer um post manual com um texto alusivo ao projeto e dar like num post;
   Esperar que o Utilizador A conclua a tarefa 5 e garantir que a timeline está atualizada
6. Analisar a timeline Beta;
7. Reordenar os posts da timeline Beta de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts;
8. Analisar a timeline Alfa;
9. Reordenar os posts da timeline Alfa de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts; (não se preocupe em assegurar a mesma ordem nos dois casos);
   Esperar que o Utilizador A conclua a tarefa 9
10. Editar o descritor “abstract” do ficheiro “Template Contrato”;
11. Partilhar um post com um frase alusiva à partilha;
    Esperar que o Utilizador A conclua a tarefa 11 e garantir que a timeline está atualizada

12. Fazer um comentário num post;
    Esperar que o Utilizador A conclua a tarefa 12 e garantir que a timeline está atualizada
13. Analisar a timeline Alfa;
14. Reordenar os posts da timeline Alfa de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts;
15. Analisar a timeline Beta;
16. Reordenar os posts da timeline Beta de acordo com a sua preferência, utilizando as setas no canto superior direito dos posts; (não se preocupe em assegurar a mesma ordem nos dois casos).

Questionário pós-tarefas

https://goo.gl/forms/UCxoHek5f433kT3K2

Appendix B

Questionnaires

B.1 Initial questionnaire

Social Dendro: questionário inicial

*Obrigatório

1. **Ocupação** *
   Marcar apenas uma oval.

   - ◯ Estudante
   - ◯ Investigador
   - ◯ Professor
   - ◯ Outra: _____

2. **Idade** *

   _____

3. **Sexo** *
   Marcar apenas uma oval.

   - ◯ Masculino
   - ◯ Feminino

4. **Classifique a frequência com que utiliza a plataforma Dendro** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Nula	◯	◯	◯	◯	◯	Elevada

5. **Classifique a frequência com que utiliza redes sociais** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Nula	◯	◯	◯	◯	◯	Elevada

6. **Indique o seu grau de experiência na área de gestão de dados de investigação** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Nulo	◯	◯	◯	◯	◯	Elevado

B.2 Final questionnaire

Social Dendro: questionário final

*Obrigatório

Classifique a importância

Escala:
1 - Nada importante
5 - Muito importante

1. **Classifique os seguintes elementos de acordo com a sua importância para a valorização de um post** *

   Marcar apenas uma oval por linha.

	1	2	3	4	5
Like					
Comment					
Share					
Data de criação					

2. **Classifique a importância/influência de cada tipo de post para a perceção da atividade de um projeto** *

   Marcar apenas uma oval por linha.

	1	2	3	4	5
Manual					
Modificação de metadados					
Upload de um ficheiro					
Eliminação de um ficheiro					
Criação de um diretório					
Eliminação de um diretório					

Concorda com as seguintes afirmações?

Escala:
1 - Discordo totalmente
5 - Concordo totalmente

3. **A timeline Alfa...** *

Marcar apenas uma oval por linha.

	1	2	3	4	5
Mostra-me informação interessante					
Mostra-me informação interessante na ordem pretendida					
Permite-me perceber rapidamente a atividade do projeto					
Permite uma maior interação entre os colaboradores do projeto					
Tem posts que podiam ser omitidos					

4. **A timeline Beta...** *

Marcar apenas uma oval por linha.

	1	2	3	4	5
Mostra-me informação interessante					
Mostra-me informação interessante na ordem pretendida					
Permite-me perceber rapidamente a atividade do projeto					
Permite uma maior interação entre os colaboradores do projeto					
Tem posts que podiam ser omitidos					

5. **Numa timeline, apresentar os posts mais interessantes primeiro leva a um maior grau de interação com a mesma.** *

Marcar apenas uma oval.

1	2	3	4	5

Resposta aberta (opcional)

6. **De que forma pode a experiência do utilizador com a timeline ser melhorada? Caso tenha uma sugestão, indique-a no espaço em baixo.**