Unexpected Errors from Metadata OAI-PMH Providers

Sarantos Kapidakis
Laboratory on Digital Libraries and Electronic Publishing, Department of Archive, Library and Museum Sciences, Ionian University, Greece

Abstract: We study the behavior and the failure reason of metadata harvesting services. We used existing OAI services and we created our own OAI client to issue requests to them for many harvesting rounds, collecting the appropriate information. We studied 1407537 harvesting tasks from 3446 harvesting services in 552 harvesting rounds during a period of 2 years, of which 618812 (44%) failed and the remaining tasks occasionally returning fewer records. We examined the reported outcome messages, the number of records returned and the response time to discover failing patterns. While most messages indicate temporary errors, we revealed messages with specific details that indicate permanent affect or no effect to the returned metadata records.

Keywords: Libraries, Open Archive Initiative, metadata harvesting, reliability, services, temporary error, permanent error, timeout, failure, quality.

1. Introduction

An established protocol for exchanging metadata is the Open Archive Initiative Protocol for Metadata Harvesting (OAI). All metadata providers act as OAI servers that accept requests to provide their metadata. A central node acts as OAI client that issues requests to many OAI servers and uses the collected metadata records to construct a central repository, which will be a new source for searching. The central node will also regularly update its metadata records with new and changed one, and therefore the OAI communication should be repeated regularly, with a new task each time. The regular tasks that request metadata records run unattended, and the system administrators assume they are successful most of the time.

Metadata harvesting is used very often, to incorporate the resources of small or big providers to large collections. The metadata harvesters, like Science Digital Library and Europeana, accumulate metadata from many collections (or sources) through the appropriate services, belonging to metadata providers, mostly memory institutions, by automatically contacting their services and

Received: 2.5.2018 / Accepted: 28.5.2018 ISSN 2241-1925

storing the retrieved metadata locally. Their goal is to enable searching on the huge quantity of heterogeneous content, using only their locally stored content. Metadata harvesting is very common nowadays and is based on the Open Archives Initiative Protocol for Metadata Harvesting. As examples, we mention the Directory of Open Access Repositories (OpenDOAR) that provides an authoritative directory of academic open access repositories, and the OAIster database of OCLC with millions of digital resources from thousands of providers.

Computer services are normally assumed to work well all the time. If a small number of harvesting tasks fails occasionally, probably due to temporary network errors, they only affect the central node temporarily – because the same records will normally be retrieved on the following harvesting tasks. The harvesting mechanism is normally established and then scheduled to run forever, but after some time we observe that a big percentage of these services stop working permanently. Kapidakis (2016a) examines how solid the metadata harvesting procedure is, by making 17 harvesting rounds, over three years, from 2014 to 2016, and exploits the results to conclude on the quality of their metadata as well as on their availability, and how it evolves over these harvesting rounds. The list of working services was decreasing every month almost constantly, and less than half of the initial services continued working at the end. The reliability of the OAI-PMH service proved to be a very important factor of the harvesting procedure.

Kapidakis (2016b) explored the behavior of the information services through frequent harvesting tasks over a small period of time so that no permanent changes to their behavior were expected. He classified the services into five classes, according to their reliability in their behaviour, and examined each class separately. He found that the service failures are quite a lot, and many unexpected situations are formed. Kapidakis (2018) analysed the message distribution per round and service to characterize them as permanent or temporary and he excluded the tasks with the temporary messages and analysed the others for more permanent characteristics.

The success or the failure of a harvesting task is often not obvious, as each task includes many stages of information exchange, and each one of them may fail – but with different consequences each time. Furthermore, tasks that complete with an error outcome message do sometimes return records and also tasks that declare that they complete normally sometimes return less than the requested or no records. Kapidakis (2017) concluded that, when we had to briefly characterise the complex procedures of a task as successful or not, we will call it successful if it returned any records. We still consider responses with fewer than the requested records as successful, as the remaining of the requested records can, hopefully, be returned on a successive request, in the common OAI-PMH configuration.

Nevertheless, the unreliability - downtime of the metadata harvesting services usually indicate a proportional unreliability or downtime of the resource providing service, which always resides on the local sites, where both the local and the harvested metadata link to. When the resources are not available, the corresponding user requests are not satisfied, affecting the quality of the service. Ward (2003) describes how the Dublin Core is used by 100 Data Providers registered with the Open Archives Initiative and shows that is not used to its fullest extent. Kapidakis (2012) presents quality metrics and a measurement tool, and applied them to compare the quality in Europeana and other collections, that are using the OAI-PMH protocol to aggregate metadata. Kapidakis (2015) further studies the responsiveness of the same OAI services, and the evolution of the metadata quality over 3 harvesting rounds between 2011 and 2013.

From the different aspects of the quality of digital library services, the quality of the metadata has been mostly studied. Some approaches are applied on OAI-PMH aggregated metadata: Bui and Park (2006) provide quality assessments for the National Science Digital Library metadata repository, studying the uneven distribution of the one million records and the number of occurrences of each Dublin Core element in these. Hughes (2005) applied another approach of metadata quality evaluation to the open language archives community (OLAC), that is using many OLAC controlled vocabularies. Ochoa and Duval (2009) perform automatic evaluation of metadata quality in digital repositories for the ARIADNE project, using humans to review the quality metric for the metadata that was based on textual information content metric values.

In this work we examined the number of returned records and the outcome messages of successful harvesting task and we found clues that will help predicting the consistency of the behavior. To do that, we gathered a lot of information by performing a large number of harvesting rounds and examined in detail the harvesting results and warning messages.
The rest of the paper is organized as follows: In section 2 we describe our methodology and how we used the software we made to create our dataset and we study characteristics of the services. In section 3 we examine the outcome of the successful and failed tasks, and in section 4 we separate the services into classes, according to the records that their tasks return, and examine their successful tasks to reveal existing patterns. We conclude on section 5.

2. Methodology, Metadata Harvesting and the Services

It is difficult to understand, analyse or predict the behavior of network services, because it depends on many factors, many of which may be external to the service and unknown. Nevertheless, there may be some significant factors of the service configuration or maintenance, or their environment (including the accessing network) that can be considered. The large number of such services, the huge amount of harvested information and the possibility of meeting transient conditions makes any monitoring and predicting a hard job.

To study the reliability of network services, we created an OAI client using the oaipy library and used it over several harvesting rounds, where on each one we asked each service from a list of OAI-PMH services for a similar task: to provide 1000 (valid – non deleted) metadata records. Such tasks are common for the OAI-PMH services, which periodically satisfy harvesting requests for the new or updated records, and involve the exchange of many OAI requests and responses, starting from a negotiation phase for the supported OAI features of the two sides.

The sources listed in the official OAI-PMH Registered Data Providers site were used as the list of our services. We started with a list of 2138 services, as on January of 2016, and (on April of 2017) we continued with an updated list of 2870 services. The two lists had 1562 entries in common, while the first list included 576 entries that were seized later on, when 1308 new entries were added. As Kapidakis (2016a, 2016b, 2017) has shown, some services (on the average) stop working every time, thus a regular update on the list of services should be in order. The update helped reducing the amount of failed tasks and increasing the services we study.

Our sequential execution of all these record harvesting tasks from the corresponding specific services normally takes much more than 24 hours to complete. Sometimes the tasks time out resulting to abnormal termination of the task: we set a timeout deadline to 1 hour for each task, and interrupted any incomplete task just afterwards, so that it will not last forever. Figure 1 show the distribution of the timeouts to the different services. Therefore, 3024 tasks had no timeouts (and are not shown) and from the 422 tasks with timeouts, 264 had one timeout, 56 had 2 timeouts and so on. The first number indicate random timeout reasons. Some services, though, had a very high timeout count, like 70, 64 and so on, and indicate a problem on the service side.

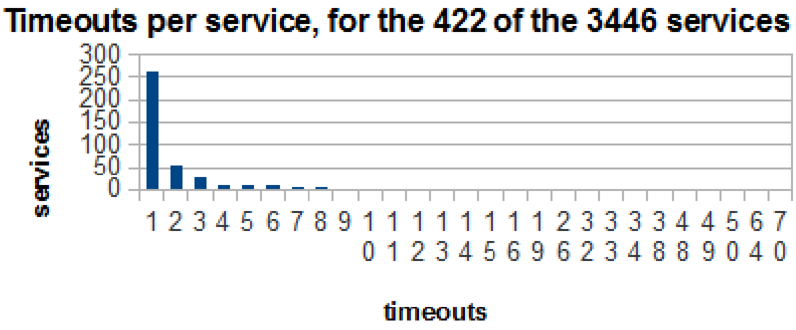

Figure 1: Number of services that timeout

We repeat a new harvesting round with a task for each service in constant intervals (ranging from 24 to 72 hours), asking the exact same requests. Each of the 3446 services was queried on 408 rounds on average (between 206 and 552). 1165 of the services always failed as none of their tasks ever returned any records, and some of them were replaced during the service list update. The behaviour of the remaining 2281 services, that do return records sometimes, is not always very consistent. Only 300 services returned records on all their tasks, and 84 of them always returned the same number of records.

Ideally, a task will complete normally, returning all 1000 requested metadata records. A task may return less records, or even 0. Additionally, a task may not declare a normal completion, but report a warning message indicating a problem, with some supplemental information detail. These two situations are not mutually exclusive: a task may declare normal completion and return no records, or a task may report a warning message and still return records – sometimes even 1000 records!

Table 1 presents information on the response time of the tasks. The maximum response time (column 3) is close to the timeout period. Successful tasks complete on average (column 4) on 54.77 seconds failed tasks (with less work done) on 30.67, much lower that the timeout period. The standard deviation is also high, indicating that accurate predictions are not possible.

Table 1: Response time of successful and failed tasks

tasks	**min**	**max**	**average**	**sdev**
success	0.39	3600.09	54.77	159.11
failure	0.00	3634.71	30.67	176.39

Table 2 presents information on performance characteristics in the behaviour of the services. The high standard deviation (last column) shows that the services do not converge to their average behaviour. From the first row we see that the tasks are expected to complete long before our timeout interval.

Table 2: Characteristics (min/max/average/sdev) for the 3446 Services

Service behavior characteristics	**min**	**max**	**average**	**sdev**
Average service response time	0.04	2270.35	42.83	101.17
Maximum service response time	0.28	3634.71	634.19	1165.19
Average number of returned records	0	1000	300.22	379.13
Number of tasks returning no records	0	552	179.57	200.02
Number of tasks returning records	0	552	228.88	200.79
Number tasks with Timeouts	0	70	0.40	2.75

3. Task Failures and Outcome Messages

Table 3 shows the outcome messages of all harvesting tasks. We can see that the failed tasks (column 5, 618812) are about as many as the successful tasks (column 4, 788725). The failed tasks may result to any of the 18 outcome messages, while the successful tasks tasks can result any of the 16 and not “SSLError” or “UnknownError” - which are not very frequent anyway.

Therefore, we cannot make assumptions based on what the outcome message is and it is not clear how severe is the issue that the outcome message indicates.

Table 3: Information and occurrences of the outcome messages

Outcome Message	**Ndetails**	**Nservices**	**Success**	**Failure**	**Total**
OK	1	2349	765074	26154	791228
URLError	12	2091	414	235811	236225
HTTPError	117	1648	3508	125011	128519
BadVerbError	42	590	107	122526	122633
XMLSyntaxError	4013	648	11926	50972	62898
NoRecordsMatchError	20	102	270	14392	14662
error	3	594	244	12297	12541
DatestampError	1	32	996	10425	11421
Error	1	152	10	7331	7341
BadArgumentError	12	24	155	6422	6577
UnicodeEncodeError	26	14	1797	2944	4741
BadStatusLine	4	244	1868	1240	3108
BadResumptionTokenError	832	22	1585	559	2144
Timeout	1	422	503	868	1371
CannotDisseminateFormatError	5	6	2	1039	1041

SSLError	1	12	0	659	659
IncompleteRead	279	123	266	161	427
UnknownError	1	1	0	1	1
Total	*5371*		*788725*	*618812*	*1407537*

Column 3 shows the number of services that any of their tasks ended up with the specific outcome message. We observe that some outcome messages occur on many tasks of the service and are not evenly distributed to nearly all services, and should depend, at a high degree, on the specific service.

Column 2 presents the number of different verbose explanation texts (which we call details) that supplement each outcome message providing more details about it. For example, the message “error” can be accompanied by one of the three explanations: “[Errno 104] Connection reset by peer”, “[Errno 110] Connection timed out” or “[Errno 113] No route to host”. Other messages, like “IncompleteRead”, may have an unlimited number of explanations,e.g. “IncompleteRead(8125 bytes read, 67 more expected)”, “IncompleteRead(8160 bytes read)” and “IncompleteRead(8184 bytes read)”.

Exactly 6 of the outcome messages are accompanied with always the same details (column 2 contains “1”), while few others have a very small number of messages, further classifying the situation to one of these cases. The remaining half outcome messages (and mostly “XMLSyntaxError”) provide many different detail messages, which include the place of the data that they refer to (e.g. where the read stopped).

4. Returned Records and Outcome Messages of Successful Tasks

In order to learn more for the reasons and the patterns of task failures, we explore the outcome messages of the successful tasks. We divide the services into 3 classes that have different return records patterns: The services in the class “full” always return all 1000 requested records, and are never affected by other factors – any outcome messages do not have any significant effect. The services in the class “less” always return the same number of records, which is less than 1000, and either they contain less records or they are affected by permanent factors. Finally, the services in the class “vary” do not always return the same number of records, and are probably affected by temporary factors.

The more harvesting rounds we use, the more services move to the "vary" class,because of temporary conditions. Nevertheless, the other classes show more stable behavior.

Table 4 shows the number of successful tasks from the services of each class and for each outcome message and each class, the number of services that these tasks belong as well as the number of different details that the supplement these outcome messages.

Table 4 reveals that, excluding the "OK", only 2 outcomes ("URLError" and "HTTPError") may not affect the number of returned records, and these occurred in exactly one successful task each. Additionally, "XMLSyntaxError" and "UnicodeEncodeError" seem to permanently affect the number of returned records, been triggered by a persistent problem. They occurred many times (206 or 302) in only one service each and seem to be related to a problem in the exchanged data (in the XML formatting or in the character encoding). When this problem is met, no more records were retrieved or exchanged. In all these cases, the outcome is supplemented with only 1 or 2 different details, which is a very small detail subset (see Table 3). All other outcomes temporarily affect the number of returned records.

Table 4: Number of successful tasks and number of services and details for each service class (full, less, vary), per outcome message

Outcome Message	full			less			vary		
	Tasks	Srv	Det	Task	Srv	Det	Tasks	Serv.	Det.
OK	28986	66	1	7260	16	1	728828	2150	1
URLError	1	1	1				413	176	7
HTTPError	1	1	1				3507	303	21
XMLSyntaxError				206	1	1	11720	103	4009
UnicodeEncodeError				302	1	2	1495	4	9
error							244	116	2
Timeout							503	181	1
DatestampError							996	7	1

IncompleteRead							266	78	166
NoRecordsMatchError							270	10	6
BadStatusLine							1868	93	2
BadResumptionTokenError							1585	19	830
Error							10	4	1
BadArgumentError							155	3	3
CannotDisseminateFormatError							2	1	1
BadVerbError							107	3	3

5. Conclusions

Communication of user and services through networking is complex, and we often do not have a picture of all the involved factors and situations. Understanding how the services behave, either qualitatively or quantitatively, help us understand the current situation and better design future services.

We used existing OAI services and we created our own OAI client to issue requests to them for many harvesting rounds, collecting the appropriate information. The large number of such services, the huge amount of harvested information and the possibility of meeting transient conditions makes any monitoring and predicting a hard job. We collected data from 1407537 harvesting tasks belonging to 3446 harvesting services in 552 harvesting rounds during a period of 2 years, of which 788725 succeeded, occasionally returning fewer records. A significant part of the OAI services never worked or have ceased working while many other services occasionally fail to respond. About half of the harvesting tasks do fail.

We demonstrated a methodology that can be used to massively examine all outcome messages (and their details) and their consequences for the harvesting tasks.

Although most of the OAI servers we queried are using a few implementations and variations of the OAI software (as can be seen from the similarity of their warning outputs), the currently used warning messages do not provide enough information on the severeness or the consequences of the warnings.

We can derive many interesting conclusions by closely examining the data in our tables. For example, although the outcome messages occur on many different situations, the success of the tasks is not affected by few specific outcomes and details, while few other outcomes and details prevent all records to be returned. Few such messages are described in section 4.

One important conclusion is that better, more informative and more uniform, warnings should be used by all OAI implementations, to better alert their administration on the needed actions that they should take.

Our results do not indicate a new approach to harvesting or conclude to a breakthrough advice, but make clear the complexity of the operation in an ever changing networking environment and alarm the reader that some facts that may be considered trivial, actually they are not! They help us to better understand the risks involved, and to design more reliable procedures and improved ways to closely monitor them.

References

Bui, Y. & Park, J., (2005). An assessment of metadata quality: a case study of the National Science Digital Library Metadata Repository. In Haidar Moukdad (Ed.) CAIS/ACSI 2006 Information Science Revisited: Approaches to Innovation. Proceedings of the 2005 annual conference of the Canadian Association for Information Science held with the Congress of the Social Sciences and Humanities of Canada at York University, Toronto, Ontario.

Hughes, B., (2005). Metadata quality evaluation: experience from the open language archives community. Berlin Springer. Lecture Notes in Computer Science vol. 3334. ISBN 978-3-540-24030-3. doi: 10.1007/b104284.

Kapidakis, S., (2012). Comparing Metadata Quality in the Europeana Context. Proceedings of the 5th ACM international conference on PErvasive Technologies Related to Assistive Environments (PETRA 2012), Heraklion, Greece, June 6-8 2012, ACM International Conference Proceeding Series; vol. 661.

Kapidakis, S., (2015). Rating Quality in Metadata Harvesting. Proceedings of the 8th ACM international conference on PErvasive Technologies Related to Assistive Environments (PETRA 2015), Corfu, Greece, July 1-3 2015, ACM International Conference Proceeding Series; ISBN 978-1-4503-3452-5.

Kapidakis, S., (2016a). Exploring Metadata Providers Reliability and Update Behavior. Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL 2016), LNCS 9819, Springer, Hannover, Germany, September 5-9, 2016.

Kapidakis, S., (2016b). Exploring the Consistent behavior of Information Services. 20th International Conference on Circuits, Systems, Communications and Computers (CSCC 2016), Corfu, July 14-17, 2016.

Kapidakis, S., (2017). When a Metadata Provider Task is Successful. Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL 2017), LNCS 10450, Springer, Thessaloniki, Greece, September 18–21, 2017, pp. 544-552

Kapidakis, S., (2018). Error Analysis on Harvesting Data over the Internet, 11th ACM International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2018, Corfu, June 26–29, 2018.

Ochoa, X. & Duval, E., (2009). Automatic evaluation of metadata quality in digital repositories. International Journal on Digital Libraries, vol. 10(2/3), pp. 67–91.

Ward, J., (2003). A quantitative analysis of unqualified dublin core metadata element set usage within data providers registered with the open archives initiative. Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries (JCDL 03), ISBN:0-7695-1939-3, pp. 315-317