When a Metadata Provider Task Is Successful

Sarantos Kapidakis(✉)

Laboratory on Digital Libraries and Electronic Publishing,
Department of Archive, Library and Museum Sciences,
Ionian University, 72, Ioannou Theotoki Street, 49100 Corfu, Greece
sarantos@ionio.gr

Abstract. Computer services are normally assumed to work well all the time. In this work we examined the operation and the errors of metadata harvesting services and tried to find clues that will help predicting the consistency of the behavior and the quality of the harvesting. The large number of such services, the huge amount of harvested information and the possibility of meeting transient conditions makes this work hard. We studied 395530 harvesting tasks from 2138 harvesting services in 185 harvesting rounds during a period of 9 months, of which 214163 ended with error messages and the remaining tasks occasionally returning fewer records. A significant part of the OAI services never worked or have ceased working while many other serves occasionally fail to respond. It is not trivial to decide when a tasks is successful, as tasks that return without an error message do sometimes return records and also tasks that declare that complete normally sometimes return less or no records. This issue is fundamental for further analysis of the harvesting outcome and any assessment that may follow. Therefore, on this work we studied the error messages and the task outcome patterns in which they appear and also the tasks that returned no records, to decide on which is the most essential condition to decide when a task is successful. Our conclusion is that a task should be considered successful when it returns some records.

Keywords: OAI · Metadata harvesting · Reliability · Services · Temporary error · Permanent error

1 Introduction

Computer services like metadata harvesting and document retrieval do not always have the expected behavior. The users may notice delays or unavailability – but they have no idea how often these happen, and assume that each problem is rare and temporary. The big diversity of computer services, their different requirements, designs and interfaces and also network problems and user-side malfunctions, make it hard to know when the behavior of a service is normal or not and what to measure in each case. To overcome some of these restrictions, we examined the behavior of a large number of similar services of a specific type: data providers using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).

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

J. Kamps et al. (Eds.): TPDL 2017, LNCS 10450, pp. 544–552, 2017.
DOI: 10.1007/978-3-319-67008-9_44

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 storing the retrieved metadata locally. Their goal is to enable searching on the huge quantity of heterogeneous content, using only their locally store content.

The reliability of the services is important for ensuring current information. If the metadata harvesting service is not responding, the corresponding metadata records will not be updated at that time. Additionally, 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.

In [5] Lagoze et al. discuss the NSDL development and explains why OAI-PMH based systems are not relatively easy to automate and administer with low people cost, as one would expect from the simplicity of the technology. It is interesting to investigate the deficiencies of the procedure.

The behavior and the reliability of a service, as well as the quality of its content, is important to the outcome and the satisfaction from the service. The evaluation and quality of metadata is examined as one dimension of the digital library evaluation frameworks and systems in the related literature, like [1, 6, 8]. Fuhr et al. in [1] propose a quality framework for digital libraries that deal with quality parameters. The service reliability falls under their System quality component.

In [7] Ward 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. In [2] Kapidakis studies the responsiveness of OAI services, and the evolution of the metadata quality over 3 harvesting rounds between 2011 and 2013. In [3] Kapidakis examines how solid the metadata harvesting procedure is, by making 17 harvesting rounds, over three years, from 2014 to 2016, and exploiting 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.

Nevertheless, the OAI services did not have a solid behavior, which make any assumption and conclusion harder to reach. In [4] Kapidakis explored the behavior of the information services over a small period of time, so that no permanent changes to their behavior were expected: during 21 harvesting rounds in three days intervals for a period of two months. He classified the services into five classes, according to their reliability in their behavior, and examined each class separately. He found that the service failures are quite a lot, and many unexpected situations are formed.

We have to address a fundamental issue: when a harvesting task should be considered successful. This issue is not trivial because (a) each harvesting task is a complex process, with many points of failure – not all of them of equal importance and (b) each harvesting task produces an outcome, that is either normal completion or an error message, but this outcome does not coincide with other task characteristics like the number of records returned: tasks that return without an error message do sometimes return records and also tasks that declare that complete without an error message sometimes return no records.

Therefore, on this work we studied the error messages and the patterns in which they appear and also the tasks that returned no records, to decide which is the most essential condition to characterise task as successful. For this reason, we performed and used a large number of harvesting rounds and examined in detail all harvesting error messages, to better understand them and the causes that triggered them.

The rest of the paper is organised as follows: In Sect. 2 we describe our methodology and the data we collected for that purpose. In Sect. 3 we study the error messages that the failed harvesting services returned and try to discover what they represent. In Sect. 4 we try to detect when the harvesting tasks time out and/or return records, to find patterns of behavior. We conclude on Sect. 5.

2 Methodology and Collected Data

It is difficult to understand, analyze 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.

To reveal common behavior patterns, we created an OAI client using the oaipy library and used several harvesting rounds, where on each one we asked each of the 2138 OAI-PMH sources listed in the official OAI Registered Data Providers (https://www.openarchives.org/Register/BrowseSites) on January of 2016 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 requests and responses, starting from the a negotiation phase for the supported features of the two sides. Our sequential execution of all these record harvesting tasks from the corresponding specific services usually takes more then a day to complete. Sometimes the tasks time out resulting to abnormal termination of the task: we set a timeout deadline to 1 h for each task.

We repeated a new harvesting round with a task for each service in constant intervals, asking the exact same requests every 36 h for a period of 9 months (June of 2016 to March of 2017). In the following, we further analyze the errors as permanent or temporary, and examine the error distribution per service.

Ideally, a task will complete normally, returning all requested metadata records – 1000, or all available records (even zero) if fewer are only available on that service. Other behaviors are also possible - and of interest when studying the behavior of each service. A task may return an error, declaring that the final task goal cannot be satisfied. We also consider as error the situation of the abnormal termination of a timed out task and such a task may return some records.

Finally, it is also possible to have a situation that the service actually returns less records than the ones available and requested, but reports that the task completes normally.

3 The Errors of the Services and Their Semantics

Our 214163 (54%) harvesting tasks did not complete normally. The reasons for such failures may be attributed to either the intermediate network or to the service – and can be either temporary or permanent. The fact that in all 185 rounds the 30155 tasks of 163 services consistently completed normally leads as to believe that the network connectivity of our OAI client is not a primary reason for the task failures in the remaining services, although we cannot exclude it, as the harvesting expands over many hours of operation.

A significant number of tasks fail on each round, and this rate is slowly increasing. The average number of failed tasks per round is 1158, with minimum 1108 and maximum 1222. Some rounds have an instantaneous increase to the number of failed tasks (round 175 is the maximum), but not too high to indicate a network problem on the OAI client side.

Our data confirms that the failed tasks are already surprisingly many and increase slowly, as was first discovered in [3]. Figure 1a shows how these task failures are mapped to the harvesting services, as each service includes 185 of these tasks, one for each round. We observe that 253 services (shown first) had no failures at all, the 1046 (shown last) services failed on all 185 of their tasks, and the remaining 839 services failed some times, most of them only a few, but others much more. Therefore, we have good, reliable services, a lot of constantly failing services and many that their outcome is affected by the environment.

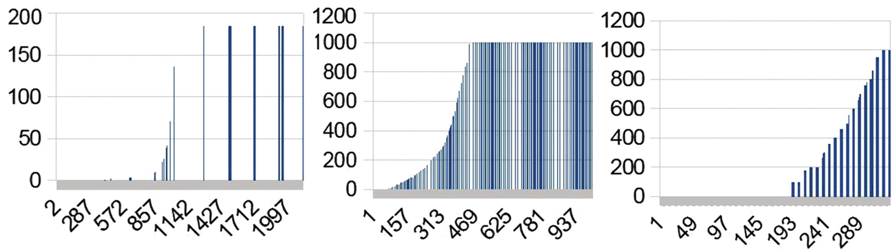

Fig. 1. (a) The number of failed tasks per service for the 2138 services. (b) The distribution of the number of records returned by a service and its frequency for the 1092 services. (c) The distribution of the number of returned records in 334 tasks that timed out.

Each OAI task consists of many actions, including mainly a negotiation in communication and many record requests. Therefore, its outcome may not be trivially classified as full success or failure, but may be something in between, with contradicting indicators, like when retrieving records and getting an error status, or when retrieving no records and completing normally. Therefore, the success of a task has to be clearly defined, so that it can be used later on. In this work, we decided to initially adopt the status returned from the task in order to characterise a success or failure, not considering the number of records actually returned. This is what each task declares anyway. This approach has the advantage that on the failures we can interpret the returned error status to further explore the reason for the failure.

The error messages reported by the failed tasks, as returned by each service and processed by the client, are briefly listed in the first column of Table 1, with the number of tasks they report them in the third column. They are service specific and the exact semantics of each error message is not globally defined.

Table 1. Service errors and the number of tasks they appear, outcome patterns they are contained and services that always return only them

Error short name	Error Category	Tasks	Outcomes	Services
URLError	Communication	68941	106	316
HTTPError	Communication	67035	96	160
BadVerbError	OAI-PMH protocol	33264	44	64
XMLSyntaxError	OAI-PMH protocol	23501	55	36
NoRecordsMatchError	OAI-PMH protocol	6082	26	4
DatestampError	OAI-PMH protocol	3835	11	6
BadArgumentError	OAI-PMH protocol	3220	10	6
error	Communication	3179	68	6
Error	Communication	1302	25	2
UnicodeEncodeError	OAI-PMH protocol	1267	5	
BadResumptionTokenError	OAI-PMH protocol	632	8	1
SSLError	Communication	610	5	1
BadStatusLine	OAI-PMH protocol	565	36	
CannotDisseminateFormatError	OAI-PMH protocol	317	3	
IncompleteRead	Communication	79	17	
Timeout		334	48	
Normal Completion		181367		253
Total		**395530**	**563**	**855**

Apart from the 181367 normal responses and the timed-out responses, we present the remaining errors by splitting them into two categories, the ones related to the network communication and the ones that are specific to the OAI-PMH operation, and its data interpretation and exchange and we show it on the second column of Table 1. We observe that the communication errors, and more specifically the *HTTPError* and *URLError*, appear much more often than the OAI-PMH protocol errors.

There are services that failed on each of their 185 tasks. There seems to be no coincidence behind this: there must be permanent issues that do not permit the service to complete normally (including possible incompatibility with our client), requiring human intervention (in the data or the software) to correct them. But these issues are not revealed easily, especially when the service does not fail in the same way each time. There may be some temporary issues, possibly on some tasks only, that apply in an earlier stage than the permanent ones, and force the task to terminate prematurely, reporting one of these issues instead, and hiding the permanent issues. Thus, the analysis of each behavior is usually complex.

In order to discover the semantics of the reported errors and the issues they imply, we have to examine their presence in many failed tasks and in the formed outcome patterns, especially since we have no access to the environment of a service (like the documentation for its operation) for further investigation.

In an effort to further clarify the semantics of the error messages, we investigated the combinations of outcomes that appear in the tasks of each service: either the error-less normal completion or an error message. By counting only the existence of the outcomes in all tasks on each service, and not their order or frequency, we reached 210 distinct outcome (completion/error) pattern combinations. The 106 of them were followed by only one service, and the rest 104 matched 2 or more services. The fifth column of Table 1 lists the number of services that follow each of the 12 service patterns that consist of only a single outcome (the same error message or normal completion) in all its 185 tasks.

In the remaining 198 outcome patterns with 2 or more (up to a maximum of 7) outcomes in their 185 tasks, the most common patterns consists of the *HTTPError* and *URLError* outcomes (93 services), and also one of these two and the normal completion (87 services). The other patterns are not followed by more than 8 services. The error messages that appear in the fifth column of Table 1 appear mostly as permanent errors, preventing any task to do any more progress.

Some error messages are more common in these patterns than others. The fourth column of Table 1 displays the error messages by frequency in the remaining 198 outcome patterns, a total of 563 error messages. There are 5 error messages that do not appear in the fourth column of Table 1: *CannotDisseminateFormatError*, *UnicodeEncodeError*, *BadStatusLine*, *IncompleteRead* and *Timeout*. These have a higher probability that they are temporary errors, but we cannot conclude that by their participation in the outcome patterns. Only *IncompleteRead* and *Timeout* are found with certainty to be temporary.

No error message seems to be part of very few error patterns. The frequency of all error messages in the patterns is more or less proportional to their overall frequency in all tasks.

4 Collective Behavior of the Services

Below, we examine the behavior of the participating services collectively, to better understand the possible outcomes, and their likelihood.

The 181367 tasks (46%) completed normally with average response time 56.9 s, although 69 of them returned no records. For tasks that returned records, the minimum response time was less than a second and the maximum was 3584 s, in a case that the information service returned 1000 records. Because of the short interval between our harvesting rounds, in most cases neither the service software nor the records in each service were modified between rounds.

The number of services that competed normally on each of the 185 rounds range from 916 to 1030, with an average of 980 and standard deviation 19. As there are no rounds that had much higher or lower normally completed task rates, we believe that there were no special conditions (possibly on our network) that affect the outcome of our tasks and contaminate our analysis, that should be excluded.

We study the distribution of the number of records returned by each service for common values, other than the requested records, 1000. Figure 1b presents the distribution of the number of records returned by each of the 1092 services that completed normally on each of the 185 rounds. 839 of the services completed normally only on some rounds, and 90 of them completed normally on all rounds, but did not provide the same number of records on each round. In these cases, the maximum number of records returned was considered. The remaining 163 services always returned the same number records. The more rounds a service completed normally – especially when their maximum is repeated in many rounds, the safer it is to assume that we have found the number of records that should be returned on each task.

Only 646 of the 1092 services returned 1000 records, and may actually hold more records. On the other hand, 69 services completed normally although always returned 0 records. The remaining 377 services returned from 2 to 991 records (270 of them returned the same records consistently in all 185 rounds – the others occasionally returned less) and with no obvious pattern. The number of returned records seems to vary unpredictably, so it should be the number of records that the service can provide, and not a number dictated by the software API..

The other 214163 (54%) tasks ended with an error: in 206138 (96%) of these tasks, no records were returned at all, while in the remaining 8025 tasks some records were indeed returned.

Among the failed tasks, 334 tasks did not terminate normally but timed out after an hour of trying. With an average service response time less than 57 s, and a standard deviation close to 137 s in the 181367 tasks that did not completed normally, our 3600 s time out deadline seems reasonable. The 188 of the timed out tasks (56%) returned no records, while 14 such tasks returned all 1000 records, but afterwards did not terminate normally, but timed out. The remaining 132 timed out tasks returned from 50 to 998 records, with an average of 442 but with no obvious concentration of values and with very high standard deviation (268).

The tasks that timed out and returned zero records are between 1 and 12 (1.7 on average) per round from the 188 tasks on 110 rounds, while the tasks that timed out and returned some records but not all 1000 are between 1 and 4 (1.3 on average) per round from the 132 tasks on 100 rounds. It seems that the tasks that timed out and returned zero records coexist more often, and the network conditions on the harvesting round have a small correlation to the time out. Figure 1c shows how the distribution of the number of returned records for the 334 timed out tasks.

Only 121 of the 2138 services ever had a time out, up to 43 ones, with an average of 2.8 and standard deviation of 6.4, in the 334 total tasks. Only 7 services timed out in more than 5 rounds, while most services only timed out in one (90 services) or two (17 services) rounds. Most of them return 0 records, but the rest of the timed out tasks return a number of records that has an almost uniform distribution, up to the maximum 1000. Our data also indicate that the time out behavior do change (on numbers and returned results) on each round.

The 8025 tasks that failed but retuned some records do failed with all 15 different error messages (the majority of them, 5030, failed with the *XMLSyntaxError* message).

Excluding the tasks that timed out, 7879 other tasks also returned records and 21 of them (with 4 different error messages) returned all 1000 requested records. The remaining 7858 tasks returned from 5 to 997 records. Thus the errors can occur on all stages of the service requests, and may not affect the task in an important way.

The 7879 tasks that failed, but not with a time out, and retuned some records had a maximum response time of 2374 s and an average of 51 s, which is even less that the maximum response time of 3585 s and the average of 60 s of the tasks that completed normally and also returned some records. Thus these failed tasks have a smaller response time, failing in an earlier stage.

The 8797 tasks that completed normally but returned no records, they all completed after a maximum of 930 s, and on the average on 3.5 s, and were the fastest of all tasks. Most of these tasks belong to 17 services that returned no records on all 185 rounds and on 16 services that returned no records on 184 rounds. Also, 34 services had one such task each, and fewer services had from 2 to 182 such tasks. We conclude that a few services consistently return no records, while others may do so temporarily.

5 Conclusions and Future Work

The OAI tasks were executed in multiple rounds, which were in small intervals to avoid significant changes on the services. The behavior of a service over multiple rounds of tasks is not always consistent, and often changes between two rounds, because of temporary problems.

OAI is a protocol that works unattended but needs site administration, maintenance and monitoring tools. OAI and its implementations are vulnerable to many network conditions and often return less records than those requested.

Most errors can appear as both permanent and temporary. The OAI servers and clients should use more fine grained error messages, and each one of them would cover with fewer semantics that will also indicate permanent and temporary unrecoverable issues.

The time out behavior seems to be strongly correlated to the specific service, as some services are much more prone to time out than others.

It seems appropriate to consider a task successful when it returns any records, even when it ended with an error or time out. Furthermore, the number of records returned, when less than the ones available by the service, does not seem biased for services either completing normally or with an error message. This consideration contradicts the reasonable alternative to only consider the return outcome for the task success, as the two considerations often disagree: when a task returns no records on its normal completion and when a task returns records while also returning an error message.

In the future, we plan combine the response time of the tasks and services with the conditions of the time out, of the incomplete read and of the return of zero records to see how they are related.

References

1. Fuhr, N., Tsakonas, G., Aalberg, T., Agosti, M., Hansen, P., Kapidakis, S., Klas, P., Kovács, L., Landoni, M., Micsik, A., Papatheodorou, C., Peters, C., Sølvberg, I.: Evaluation of digital libraries. Int. J. Digit. Libr. **8**(1), 21–38 (2007)
2. Kapidakis, S.: Rating quality in metadata harvesting. In: Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA 2015), Corfu, Greece, 1–3 July 2015. ACM International Conference Proceeding Series (2015). ISBN 978-1-4503-3452-5
3. Kapidakis, S.: Exploring metadata providers reliability and update behavior. In: Fuhr, N., Kovács, L., Risse, T., Nejdl, W. (eds.) TPDL 2016. LNCS, vol. 9819, pp. 417–425. Springer, Cham (2016). doi:10.1007/978-3-319-43997-6_36
4. Kapidakis, S.: Exploring the consistent behavior of information services. In: CSCC 2016, Corfu, 13–16 July (2016)
5. Lagoze, C., Krafft, D., Cornwell, T., Dushay, N., Eckstrom, D., Saylor, J.: Metadata aggregation and automated digital libraries: a retrospective on the NSDL experience. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2006), pp. 230–239 (2006)
6. Moreira, B.L., Goncalves, M.A., Laender, A.H.F., Fox, E.A.: Automatic evaluation of digital libraries with 5SQual. J. Informetrics **3**(2), 102–123 (2009)
7. Ward., J.: A quantitative analysis of unqualified dublin core metadata element set usage within data providers registered with the open archives initiative. In: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2003), pp. 315–317 (2003). ISBN 0-7695-1939-3
8. Zhang, Y.: Developing a holistic model for digital library evaluation. J. Am. Soc. Inf. Sci. Technol. **61**(1), 88–110 (2010)