Longitudinal Navigation Log data on a Large Web Domain

Suzan Verberne
Radboud University, Nijmegen, the Netherlands
s.verberne@cs.ru.nl

Bram Arends
Radboud University, Nijmegen, the Netherlands
b.j.w.t.arends@student.ru.nl

Wessel Kraaij
TNO, the Hague
Radboud University, Nijmegen, the Netherlands
Leiden Institute of Advanced Computer Science, Leiden University
kraaijw@acm.org

Arjen de Vries
Radboud University
Nijmegen, the Netherlands
arjen@acm.org

ABSTRACT

We have collected the access logs for our university's web domain over a time span of 4.5 years. We now release the pre-processed data of a 3-month period for research into user navigation behavior. We preprocessed the data so that only successful GET requests of web pages by non-bot users are kept. The resulting 3-month collection comprises 9.6M page visits (190K unique URLs) by 744K unique visitors.

CCS Concepts

•**Information systems** → **Web log analysis; Traffic analysis;** *Web searching and information discovery;*

Keywords

data collection; access logs; navigation behavior; link analysis; graph clustering

1. INTRODUCTION

Finding information on the web requires a combination of searching and browsing. Browsing a large web domain can sometimes be challenging because traditional navigation structures do not support a complex hierarchy of pages and the possibility of multiple entry points [4]. In this paper we address the navigation behavior of users on a so-called *mega-site*. Mega-sites are extremely large, many levels deep, made up of many subsections, cater to many audiences and have multiple entry points [4]. Organizations with mega-sites are typically large organizations with many divisions and multiple target groups. Universities are prototypical examples of such organizations.

We present a newly released data collection that comprises longitudinal navigation log data on a mega-site: a university web domain. The complete data set consists of 4.5 years of access data. A first set of 3 months will now be released as a resource for research. The data collection is unique in two ways: (1) its size: the pre-processed data of only 3 months comprises almost 10 Million page visits; (2) its richness: all information from the original access log entries, including URL strings, referrers and timestamps, has been preserved.

Table 1: Statistics for the data collection, showing the cumulative effect of filtering applied to the raw log files.

# of entries in raw sample	164,923,821
# of entries after success status filtering	121,840,127
# of entries after request type filtering	115,866,318
# of entries after file type filtering	12,963,852
# of entries after web bot filtering	9,633,438
# of entries after anonymization	9,555,418
# of unique users in sample	744,340
# of unique URLs in sample	190,457

The data collection is valuable for a range of research topics: user navigation, browsing and stopping behavior, evaluation of clustering techniques on large (directed) graphs, evaluation of link prediction and page recommendation techniques, clustering of web users based on their online activities, evaluation of content-based and navigation-based user profiles, and the evaluation of usability of mega-sites.

In this paper, we describe the preprocessing of the data, present some statistics of the resulting data collection and show two examples of research that can be conducted with this data collection: link prediction and graph clustering.

2. DATA

We have been collecting (and are still collecting) server access logs since September 2011 on the web domain of our university. The data is constantly growing, with a speed of more than 50 Million page visits per month. The data that we make available for research covers three months of navigation logs: from October 1st to December 31st 2014. The raw data comprises 156M page visits (1.3M unique URLs).

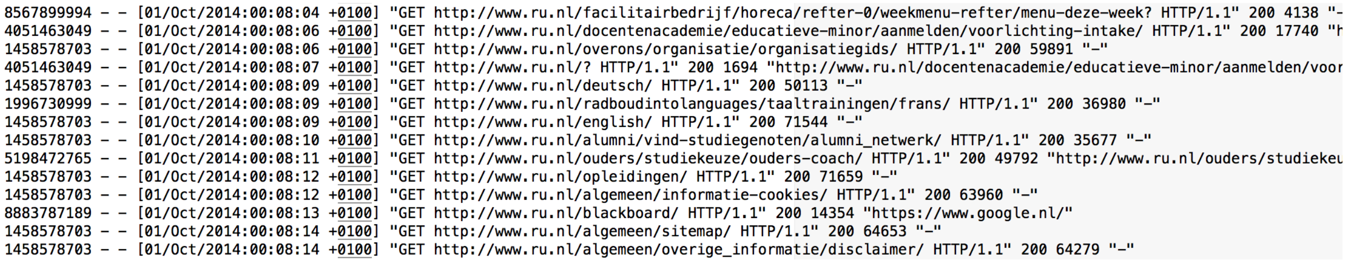
2.1 Preprocessing

The log data are formatted as Apache log files.[1] We filtered the raw data as follows: We removed all requests that did not result in a successful response (status codes starting

SIGIR '16, July 17–21, 2016, Pisa, Italy.

ACM ISBN 978-1-4503-4069-4/16/07..
DOI: http://dx.doi.org/XXXX.XXXX

[1] See https://httpd.apache.org/docs/1.3/logs.html for a definition of the format.



```
8567899994 - - [01/Oct/2014:00:08:04 +0100] "GET http://www.ru.nl/facilitairbedrijf/horeca/refter-0/weekmenu-refter/menu-deze-week? HTTP/1.1" 200 4138 "-
4051463049 - - [01/Oct/2014:00:08:06 +0100] "GET http://www.ru.nl/docentenacademie/educatieve-minor/aanmelden/voorlichting-intake/ HTTP/1.1" 200 17740 "h
1458578703 - - [01/Oct/2014:00:08:06 +0100] "GET http://www.ru.nl/overons/organisatie/organisatiegids/ HTTP/1.1" 200 59891 "-"
4051463049 - - [01/Oct/2014:00:08:07 +0100] "GET http://www.ru.nl/? HTTP/1.1" 200 1694 "http://www.ru.nl/docentenacademie/educatieve-minor/aanmelden/voor
1458578703 - - [01/Oct/2014:00:08:09 +0100] "GET http://www.ru.nl/deutsch/ HTTP/1.1" 200 50113 "-"
1996730999 - - [01/Oct/2014:00:08:09 +0100] "GET http://www.ru.nl/radboudintolanguages/taaltrainingen/frans/ HTTP/1.1" 200 36980 "-"
1458578703 - - [01/Oct/2014:00:08:09 +0100] "GET http://www.ru.nl/english/ HTTP/1.1" 200 71544 "-"
1458578703 - - [01/Oct/2014:00:08:10 +0100] "GET http://www.ru.nl/alumni/vind-studiegenoten/alumni_netwerk/ HTTP/1.1" 200 35677 "-"
5198472765 - - [01/Oct/2014:00:08:11 +0100] "GET http://www.ru.nl/ouders/studiekeuze/ouders-coach/ HTTP/1.1" 200 49792 "http://www.ru.nl/ouders/studiekeu
1458578703 - - [01/Oct/2014:00:08:12 +0100] "GET http://www.ru.nl/opleidingen/ HTTP/1.1" 200 71659 "-"
1458578703 - - [01/Oct/2014:00:08:12 +0100] "GET http://www.ru.nl/algemeen/informatie-cookies/ HTTP/1.1" 200 63960 "-"
8883787189 - - [01/Oct/2014:00:08:13 +0100] "GET http://www.ru.nl/blackboard/ HTTP/1.1" 200 14354 "https://www.google.nl/"
1458578703 - - [01/Oct/2014:00:08:14 +0100] "GET http://www.ru.nl/algemeen/sitemap/ HTTP/1.1" 200 64653 "-"
1458578703 - - [01/Oct/2014:00:08:14 +0100] "GET http://www.ru.nl/algemeen/overige_informatie/disclaimer/ HTTP/1.1" 200 64279 "-"
```

Figure 1: Screenshot of small portion of the filtered data with masked IP addresses in the first column.

with 3 or higher); all requests that are no GET requests; and all requests for images and other files that do not result from a navigational process.[2] In addition, we removed all requests that supposedly come from web bots, using the regular expression `.*(Yahoo! Slurp|bingbot|Googlebot).*` on the log entry. We anonymized the data by taking the following measures:

- We replaced all occurrences of the same IP address by a unique random identifier (a 10-digit string).
- We removed the last part of each log entry – the User-Agent HTTP request header – which is the identifying information that the client browser reports about itself.[3]
- If the referrer is a search engine, we removed everything after the substring `/search?`. We are aware that queries can provide valuable information about pages in the domain [2], but queries are also known to potentially be personally identifiable information [1]; for that reason, we will postpone a decision on releasing filtered query information, and first gain experience with the external usage of the data without search queries.
- We removed requests for URLs that only occur once in the 3-month-dataset to reduce the chance of unmasking specific users. This is an additional security step since extremely low-frequent URLs are highly specific and therefore often unique for a person.

The effect of each of the filtering steps is shown in Table 1. The information that is retained per entry is: unique user id, timestamp, GET request (URL), status code, the size of the object returned to the client, and the referrer URL. A sample of the resulting data is shown in Figure 1. The sample illustrates that the content (URLs and referrers) is multilingual: predominantly Dutch, and English and German in smaller proportions.

2.2 Statistics on the resulting collection

The final size of the collection is 9.6M page visits (190K unique URLs) by 744K unique visitors. Figure 2 shows the distribution of the number of occurrences of GET requests in the collection, on a log-log scale. The plot indicates the expected long-tail (power law) distribution: very few pages occur highly frequently, and many occur infrequently.

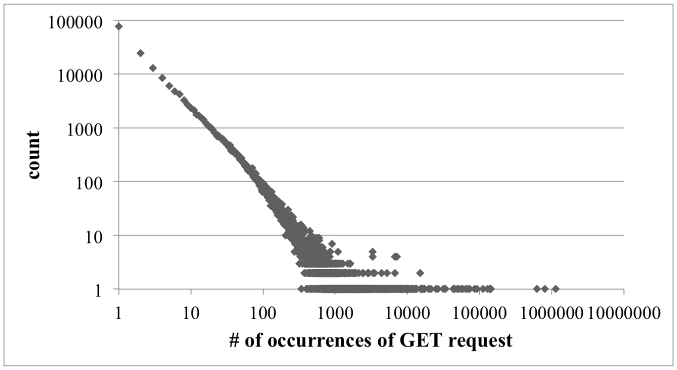


Figure 2: Distribution of the number of occurrences of unique GET requests in the collection.

2.3 Availability

The data collection is available to researchers through DANS (Data Archiving and Networked Services).[4] Users of the dataset are asked to sign a data agreement stating that they will use the dataset for research purposes, that they will not attempt to discover the identities of the persons whose navigation behavior is logged in the collection, and that they will erase the data after completion of their research.

3. EXAMPLE RESULTS

In this section, we describe the results of two experiments with the data, one addressing the task of link prediction and one addressing the task of URL clustering. Both tasks have only been evaluated before on much smaller datasets. The results presented here demonstrate the utility of the new data collection for the evaluation and validation of graph methods on large data sets.

3.1 Link prediction

Our first goal was to investigate how well we could predict the next request of a user based on the navigation behavior of other users. This task is called link prediction. Link prediction can be employed for recommending relevant pages to the user and/or webmaster and thereby improving the user experience on a web domain. We use a subset of our data for our experiments: one week of logs (7 days) were used as training data, and one day (the 8th day) was used as test data. This resulted in a training set with 35,000 unique URLs and 85,000 unique users. Although this is a

[2] All files with one of the extensions `axd`, `aspx`, `bmp`, `doc`, `docx`, `jpg`, `jpeg`, `gif`, `css`, `js`, `png`, `sav`, `swf`, `ttf`, `txt`, `xls`, `xlsx`, `xml`, `zip`, `ppt`, `pptx`, `ico`

[3] The Electronic Frontier Foundation shows the potential of this information: https://panopticlick.eff.org/

[4] http://dx.doi.org/10.17026/dans-28m-mwht

very small portion of our data, it is much larger than the data sets used in previous work on link prediction, with 100 to 1000 unique URLs [8, 11].

Method. We adopted the most commonly used method for link prediction: Markov chains [8, 11]. A Markov chain describes a system of transitions between states, where the probability of going to a state only depends on the previous state. In our model, the states represent the visited pages and the transitions represent the probability of a user visiting two pages directly after each other.

Let $s(t-1)$ be the vector representing the state during time $t-1$, and A the transition matrix with the probabilities of going from one state to another. Then the probability of going to $s(t)$ is computed with:

$$s(t) = s(t-1)A \tag{1}$$

The transition matrix A is computed with:

$$A(s, s') = \frac{C(s, s')}{\sum_{s''} C(s, s'')} \tag{2}$$

where $A(s, s')$ is the probability of going from state s to state s', and $C(s, s')$ is the number of times s preceeds s' in the training data [8]. The resulting transition matrix for our data is sparse.

We investigated the effect of the history size (using 1, 2 or 3 previously visited pages) on the quality of prediction of the next page. We compared two interpolation strategies from the literature [8, 11] for combining the previous stages:

$$s(t) = a_1 i_{t-1} A + a_2 i_{t-2} A^2 + ... \tag{summation}$$

$$s(t) = \max(a_1 i_{t-1} A,\ a_2 i_{t-2} A^2,\ ...) \tag{maximization}$$

where $s(t)$ denotes the probability of going to state s during time t, i_t is the vector representing the state, with a probability 1 at that state at that time, and a_i are user specified weights; we use uniform weights in our experiment.

Experimental setup. The requests in the training and test data were grouped by user and sorted by timestamp, resulting in a full path of requested pages per user. We divided these paths into sessions: after at least thirty minutes of inactivity a new session was started. We used sessions from 1,000 randomly selected users from the test data for evaluation; we repeated the experiment five times with a different random seed and report the average accuracy over the seeds. We vary the size of the history (1, 2 or 3 previous pages) and the interpolation strategy for previous stages (summation or maximization).

For each page in the test data that has sufficient history in its session, our algorithms try to predict the next visited page. Note that the interpolation over previous stages functions as a back-off strategy for missing links in the training data: if we try to predict the navigation sequence $A \rightarrow B \rightarrow C$, after seeing $A \rightarrow B$ in the test data, it might be that the path $A \rightarrow B \rightarrow C$ is not present in the training data, but the path $B \rightarrow C$ is. In both interpolation strategies, the probability of $B \rightarrow C$ is a factor in prediction. However, if C does not occur in the training data at all, predicting C is not possible. In our experiment, 3% of the URLs in the test data does not occur in the training data.

Results. We let the Markov chain return the page predictions in decreasing order of probability. We evaluated the prediction quality in terms of Success@1 and Success@3: the proportion of predictions with the correct next URL returned in the top-1 or the top-3 respectively. The results, presented in Table 2, show that there is no clear preference for one interpolation method over the other. If we use a history of three URLs in the session, 90% of the returned URL suggestions has the correct next URL in the top-3. This is a result that would be useful in a recommendation setting on the web domain: providing the user with three suggestions for the next URL to visit.

Table 2: Evaluation of link prediction (Success@1 and Success@3 scores, averaged over 5 random data seeds), using two strategies for combining the previous states and three history sizes (number of previous pages used).

History size	Summation		Maximization	
	S@1	S@3	S@1	S@3
1	64.3%	82.9%	64.3%	82.9%
2	46.9%	61.2%	46.7%	61.1%
3	79.4%	90.0%	80.3%	90.3%

Interestingly, the results for a history of two URLs are worse than the results for a history of one URL, even though the interpolation of previous stages ensures a back-off mechanism. We speculate that there are many sessions with two URLs – the first URL serving as history for the second – and these are relatively easy to predict, compared to the sessions with three URLs, where there is more variety in the history: If $A \rightarrow B \rightarrow C$ is not present in the training data, but $A \rightarrow B \rightarrow X$ and $A \rightarrow B \rightarrow Y$ are, with high probabilities, then the back-off probability of $B \rightarrow C$ cannot compensate for the wrong X and Y predictions.

This initial result shows the potential of link prediction on the navigation graph, and provides future directions for research into the use of Markov chains on large directed graphs, and the modeling of user navigation behavior.

3.2 Web page clustering

In a second study, we investigated the clustering of web pages based on the transitions of visitors between URLs. In previous work, techniques for clustering navigation patterns were only evaluated on smaller data sets [9, 10]. Most graph clustering techniques that can handle large graphs have been developed for undirected graphs (for community detection) [6, 5]. Spectral clustering methods are often used to cluster directed graphs, but applying them to large graphs is not feasible [5].

Method. A solution to our problem is provided by Rosvall and Bergstrom (2008) [7]. They use a random walk method to find a good clustering, by compressing the bit sequence of a random walk with Huffman coding. Huffman coding is a compression technique that assigns short codewords to common bit sequences and long codewords to rare ones; the most frequently visited nodes in a random walk get the shortest codes. The average number of bits to describe one step in the graph is a measure for the quality of the clustering. The idea is that a good clustering will result in a better compression for the bit sequence than a bad clustering, because when performing a random walk, one is more likely to stay in a good cluster for a long time before leaving. A clustering that minimizes the entropy of the bit sequence would be optimal. Finding the minimal entropy can be achieved with the Louvain algorithm, which greedily

Table 3: The first 6 URLs from the five largest clusters found by clustering the navigation paths as directed graphs.

1	http://www.ru.nl/?
	http://www.ru.nl
	http://www.ru.nl/vm_syllabus_plus/rooster
	http://www.ru.nl/algemene_onderdelen/zoeken/
	http://www.ru.nl/vacatures/details/details_v
	http://www.ru.nl/vacatures/alle-vacatures
2	http://www.ru.nl/studereninnijmegen/?
	http://www.ru.nl/studereninnijmegen/opleiding
	http://www.ru.nl/studereninnijmegen
	http://www.ru.nl/studereninnijmegen/voorlicht
	http://www.ru.nl/studereninnijmegen/vm/zoeken
	http://www.ru.nl/studereninnijmegen/opleiding
3	http://www.ru.nl/ubn/?
	http://www.ru.nl/ubn
	http://www.ru.nl/ubn/literatuur_zoeken/zoeks
	http://www.ru.nl/ubn/literatuur_zoeken/onder
	http://www.ru.nl/ubn/literatuur_zoeken/vakge
	http://www.ru.nl/ubn/literatuur_zoeken/vakge
4	http://www.ru.nl/nieuws/?
	http://www.ru.nl/nieuws
	http://www.ru.nl/nieuws/@797633/rotonde-oke
	http://www.ru.nl/nieuws/persberichten-0/vm/o
	http://www.ru.nl/nieuws/opinie/columns/colum
	http://www.ru.nl/nieuws/dossiers/dossiers
5	http://www.ru.nl/students/?
	http://www.ru.nl/students/masters/masters-pr
	http://www.ru.nl/students/masters/masters-pr
	http://www.ru.nl/students/masters/admission/
	http://www.ru.nl/students/masters/financial-
	http://www.ru.nl/students/bachelors/bachelor

minimizes the entropy in $O(n \log n)$ (with n being the number of nodes). In the case of sparse graphs, its complexity appears to be linear with the number of nodes [3].

We cluster the transition matrix A from the Markov chain, in order to get a clustering of the web pages in the data collection. The already available transition probabilities from one node to another are used to construct the random walk. One parameter that needs to be set is the teleportation probability, which is the probability to jump to a random node in the graph at any time. This is needed when the random walk gets stuck in dead ends or cycles. We set the teleportation probability to 0.15, which is the same value as in Google's PageRank algorithm and proposed in the paper by Rosvall and Bergstrom [7].

Experimental setup. We again used the sample of one week for these experiments, comprising 35,000 unique URLs. Note that we only use transitions between pages for the clustering, not the URL strings or the content of the URLs.

Results. 2,915 clusters were generated for the set of URLs in our sample. The majority of clusters contains only one or two URLs, which is probably caused by the sparsity of the graph. We manually went through the clusters and we found that the 20 largest clusters represent sensible groups of URLs. As illustration, Table 3 shows the first 8 URLs from the five largest clusters. Clusters 2, 3, 4 and 5 turn out to comprise specific subdomains. Cluster 1 (which is the largest cluster) contains general and often-visited pages.

This initial result provides future directions for research into the development of clustering techniques on large directed graphs, especially for modeling user navigation behavior, and clustering users into visitor groups.

4. CONCLUSIONS

We have prepared a data collection of user interaction logs with a university web site. The collection comprises 3 months of data, filtered for unsuccessful requests, download file types and web bots, and anonymized while preserving unique visitor IDs. The size of the preprocessed data is 9.6M page visits (190K unique URLs) by 744K unique visitors. The data collection allows for research on, among other things, user navigation, browsing and stopping behavior and web user clustering. We have conducted two exploratory studies on a one-week subset of the data, directed at link prediction and graph clustering. The results show the potential of the data, especially given its size and its richness.

For experiments that require larger data sets it will be possible to release more data in the future: we have been collecting data for 4.5 years and the raw data is growing at a speed of 50 Million page visits per month.[5] In addition, we will investigate the possibility to release the domain's query logs together with the navigation log data.

Acknowledgements

This publication was supported by the Dutch national program COMMIT (project P7 SWELL).

5. REFERENCES

[1] Adar, E.: User 4xxxxx9: Anonymizing query logs. In: Proc of Query Log Analysis Workshop, International Conference on World Wide Web. (2007)

[2] Antonellis, I., Garcia-Molina, H., Karim, J.: Tagging with queries: How and why? In: Second ACM International Conference on Web Search and Data Mining WSDM 2009, Late Breaking Results Session, Infolab (February 2009)

[3] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10) (2008) P10008

[4] Boag, P.: Navigation for mega-sites. Smashing Magazine (2013) Accessed on January 20, 2016.

[5] Fortunato, S.: Community detection in graphs. Physics Reports **486**(3) (2010) 75–174

[6] Malliaros, F.D., Vazirgiannis, M.: Clustering and community detection in directed networks: A survey. Physics Reports **533**(4) (2013) 95–142

[7] Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences **105**(4) (2008) 1118–1123

[8] Sarukkai, R.R.: Link prediction and path analysis using Markov chains. Computer Networks **33**(1) (2000) 377–386

[9] Smith, K.A., Ng, A.: Web page clustering using a self-organizing map of user navigation patterns. Decision Support Systems **35**(2) (2003) 245–256

[10] Xu, J., Liu, H.: Web user clustering analysis based on K-Means algorithm. In: Information Networking and Automation (ICINA), 2010 International Conference on. Volume 2., IEEE (2010) V2–6

[11] Zhu, J., Hong, J., Hughes, J.G.: Using Markov chains for link prediction in adaptive web sites. In: Soft-Ware 2002: Computing in an Imperfect World. Springer (2002) 60–73

[5] Please contact the authors if you are interested in obtaining a larger collection.