How to gather data using a web crawler: An application using SAS to search EDGAR

Joseph Engelberg

(Northwestern University)

Srinivasan Sankaraguruswamy*

(National University of Singapore)

***Contact Author**
Srinivasan Sankarauguruswamy
1 Biz Link
NUS Business School
National University of Singapore
Singapore 117592
bizsrini@nus.edu.sg
(65) 6516-4473

How to gather data using a web crawler: An application using SAS to search EDGAR

Empirical researchers in economics, finance and accounting grapple with a fundamental problem of gathering a representative sample of firms to test various theories that abound in the literature. Several well established data vendors provide a wealth of information toward this endeavor. However, data providers only provide a portion of information available. In this paper we provide a simple SAS program that can search for particular phrases in any Form filed by a registrant with the SEC. This allows researchers to "crawl" the web and access a large trove of data disclosed by managers in their public filings.

How to gather data using a web crawler: An application using SAS to search EDGAR

Empirical researchers in economics, finance and accounting grapple with a fundamental problem of gathering a representative sample of firms to test various theories that abound in the literature. Several well established data vendors provide a wealth of information toward this endeavor. However, data providers only provide a portion of information available. For example, COMPUSTAT provides financial statement variables found in a firm's 10-K that might constitute a few pages worth of information, but many firms have 10-Ks that are well over 50 pages long, most of which has not been codified by any data vendor. The SEC has mandated that all their registrants should file their documents online using the EDGAR system. Prior literature has gathered data using web based techniques to obtain a larger (cleaner) dataset than provided by data vendors. One such example is a paper by Butler, Leone, and Willenborg (2004). The authors in the first line of their abstract lay out the advantage of gathering a large database from the web in the following way "in this paper we use a web based sampling methodology to obtain and content analyze a large sample of modified audit opinions." In this example, a firm's 10-K is publicly available on the World Wide Web (the web from now on) so that the additional information could be obtained through a technology that systematically accesses 10-K files on the web. Another example of using the web to gather data from a 10-K can be found in Whisenant, Sankaraguruswamy and Raghunandan (2003). The authors gather audit fee data from EDGAR using search phrases using a web crawling program.

This technology has also been used in gathering data from websites that are not part of a regulatory agency. Because of an increase in publicly available information on the web, there has been a growing trend of using a "web crawling" technology by researchers to gather information vital to their research. For example, the use of web crawling has helped launch the online auction literature; by using publicly available information on sites like eBay, several authors have gathered information via web crawling to make advances in the understanding of bidding behavior and reputation effects of consumers (see Engelberg and Williams (2006), Roth and Ockenfels (2002), and Bajari and Hortacsu (2003)). More recently, Antweiler and Frank (2004) gathered data from stock message boards via web crawling to see whether the “chatter” on these boards was related to stock price returns and volatility.

The above papers gather data using web crawlers when they have had a reasonable expectation that such data was available on the web. It is also possible to search for data using a web crawler to check whether a particular potentially interesting phenomenon is wide spread or rare. One example would be searching for the use of put warrants by firms that repurchase stocks. This phenomenon is interesting in that firms seem to use this to signal their quality, however, it is not wide spread and hence gathering data manually is not easy. Gibson, Povel and Singh (2004) find that hand searching for phrases like “put warrants” or “put options” and excluding warrants and options related to commodities yielded 386 observations for 85 firms over 7 years. This kind of data collection is made easy by the use of a computer based search algorithm.

While there are many applications for a web crawling technology, we will focus on the EDGAR system that the SEC has put in place to make available all the filings of its registrants on the web. This treasure trove of information can be mined by researchers but at a cost. Searching through the myriad forms and the millions of filings is not an easy task. In this paper we provide one simple example which will allow a researcher to attack this problem using a web crawling program written for software that is widely used by researchers for other purposes, i.e. SAS.

We present several steps that are involved in using a web crawler (in SAS) efficiently, for searching data on EDGAR.

Step 1: Identify the form or filing in which the data is available.

The EDGAR database has over 605 different forms and 4,249,586 filings between 1994 and 2006. It is important to have a general idea about which forms would contain the data that we need to gather. For example, audit fees data is usually available in proxy statements (Form Def 14-A), going concern opinions and put warrants, in quarterly and annual financial statements (Forms 10-Q and 10-K), and age of the firm can be obtained from data in the Management Discussion and Analysis (MD&A). Hence a researcher should at the very least know which form would contain the data she is interested in.

Step 2: Gathering the URL for all forms filed with the SEC.

The location of every SEC form is documented in master.idx files located at: ftp://www.sec.gov/edgar/full-index/. There is a master.idx file generated for every quarter. For example, to see all forms filed in the forth quarter of 1998, go to ftp://www.sec.gov/edgar/full-index/1998/QTR4/master.idx . The contents of the master.idx files can be read in SAS and a sample program can be downloaded from: *http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/make_accession_masterlist.sas*

To make the master.idx files readable in SAS first the files should be renamed after each of them is downloaded. The naming convention that we have used in the sample SAS program is companyyyq.idx, where yy is 97 for 1997, 100 for 2000 and 101 for 2001, and q is the quarter and takes values 1 through 4. A zip file with all the master.idx files in their original text format can be found at: *http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/companyidx_textformat.zip*

To access the SAS datasets that contains the URLS please use the following links:

http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2006.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2005.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2004.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2003.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2002.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2001.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2000.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_1994to1999.zip

The URL of each of the filings is given in the master.idx file and will be used to access the form stored on the SEC website. Make a comprehensive file of all the master.idx for all the years till date. Then keep only those filings pertaining to a particular form (s) that you want to search.

Step 3a: If the superset of firms is known

In some cases the search is predicated on some filtering criteria that are not related to the web crawling on EDGAR, for example, studying share repurchases, or, NYSE firms, or, COMPUSTAT firms. First set up the superset of such firms. Next match the identification variable on the superset with the CIK (Central index Key). This matching of CIK with say PERMNO or GVKEY can be procured from S&P. If you do not have access to this match see section at the end of the paper to obtain a rough match. Then merge the set of firms already filtered with the set of filings retained in Step 2.

Step 3b: If the superset of firms is not known

In other cases the search is not predicated on any filtering criteria and the researcher wants to search all incidences of a particular phenomenon. In this case use the sample of filings retained in Step 2.

Step 4: Run the web crawling program

From Step 3 above the researcher obtains the sample of firm filing observations that the researcher is potentially interested in. Now identify the phrase that uniquely delineates the phenomenon of interest. For example, when searching for audit fees paid to an auditor, the search phrases could be “audit fees”, “fees paid to auditor”, when searching for put options the search phrases could be “warrant”, or “put options”. This needs to be input into the SAS program, which can be downloaded from: http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/webcrawler-example.sas

The example SAS program has been set up to search for the age of the firm and so we need to gather data about when the firm was founded, or organized, or incorporated. These search terms are then input into the SAS program as counters. When the program encounters any of these words the counter starts ticking. Later in the program the researcher can decide how many lines before and after the first incidence of each search term to retain and output into a SAS dataset. An example input dataset has been provided to help the researcher test the program on their system, and can be downloaded from: http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/test_age.sas7bdat

A list of the output generated from web crawling for age on this example input dataset can be found at: http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/cumulate_age.sas7bdat

Step 5: Analyzing the output.

The data of interest is output to a SAS dataset and can be exported to an EXCEL file for better readability. Since this form of data capture is not exact, the researcher will need to expend some effort to "clean up" the data for further processing. This could either be in the form of further data input or keeping the list of relevant CIK which exhibit the particular phenomenon of interest to the researcher.

CIK – CUSIP Match

In case you do not have access to the CIK-CUSIP Match provided by S&P, you can get a reasonable match using the EIN (Employer Identification Number) that is available on COMPUSTAT. The EIN can be obtained by web-crawling the FORM 10-K. To do this first get all the unique CIKs' with FORM 10-K from the SAS datasets given in the links above. Then use the web crawling program and give a generic keyword like auditor, the EIN is one of the fields that the web crawling program spits out. Then match the EIN in COMPUSTAT with the EIN obtained from the web crawling program. This yields a reasonable match between CUSIP and CIK, at least for active firms.

NAMES MATCH

If you have a dataset of names and want to match to CIK using firm names then you can do this by first crawling all Form 10-K's and getting the firm names and attached CIK. Then use the name_match.sas program that is provided in the link below and match by name. The name match is not exact but it outputs a score for each name match and you can pick a cut off below which you can hand match. It reduces your hand collection by a significant degree. There is a SAS function [SPEDIS()] which can be used instead of the name match program, which yields a spelling score. The link for the name match SAS program is

http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/name_match.sas

References

Antweiler, W., and M. Frank (2004). “Is all that talk just noise? the information content of internet stock message boards,” *Journal of Finance*, v59(3), 1259-1295.

Bajari, P. and A. Hortacsu (2003): “The Winner’s Curse, Reserve Prices and Endogenous Entry: Empirical Insights from eBay Auctions,” *Rand Journal of Economics* Vol. 3, No. 2, pp. 329-355.

Butler, M., A. Leone., and M. Willenborg (2004) “An empirical analysis of auditor reporting and its association with abnormal accruals,” *Journal of Accounting and Economics*, 37, 139-165.

Engelberg, J. and J. Williams (2006): “EBay’s proxy system: a license to shill,” Working paper, Kellogg School of Management..

Gibson, S., P. Povel, and R. Singh. (2006) “The information content of put warrant issues,” Working paper, College of William and Mary, Williamsburg.

Roth, A. and Ockenfels, A. (2002) ”Last minute bidding and the rules for ending second-price auctions: evidence from eBay and amazon on the internet” *American Economic Review* 92(4).

Whisenant, S., S. Sankaraguruswamy, and K. Raghunandan. (2003) “Evidence on the joint determination of audit and non audit fees,” *Journal of Accounting Research* 41(4), 721-744.

Appendix – Example SAS Program

```
/************** The input to the process below is a dataset name test
(you can change this in the macro call command
to what ever dataset name you like). This dataset contains as variables
cika - the central index key (Text format)
accession - the accession link from the company.idx (compiled in
dataset company_idx.sas7bdat
form - the form that you want to crawl (text format)
These variables are available in the zip files from
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/companyidx_textformat.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2006.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2005.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2004.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2003.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2002.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2001.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_2000.zip
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/company_idx_sasdataset_1994to1999.z
ip

The SAS program that creates the sas datasets is the following
http://www.bschool.nus.edu.sg/staff/bizsrini/webcrawl/make_accession_ma
sterlist.sas

**********************************************************************/

/*************** Read in the dataset ***************/
libname urllist '' ;
data test ; set urllist.all ; keep date datefile name accession cika
form ; run ;

/* Merge with the data you want to use to crawl based on CIK.
You may want to convert the cika to numeric if the cik you have is
numeric
*/

************** In this example, the webcrawler will visit a
predetermined list of websites (each an SEC form) and retrieve lines
surrounding keywords like "INCORPORATED IN", "INCORPORATED UNDER",
"FOUNDED IN", "FOUNDED UNDER", etc.  The code demonstrates SAS's
ability to crawl through thousands of SEC files and can be easily
modified to suit other kinds of search and retrieval requests
*********;



filename junk dummy;
proc printto log=junk; run;   ***** Turns Off Log File so that it does
not get full and stop the program ****;
```

```
%MACRO webcrawler (ds=);

/************* Code to determine the number of observations (i.e. -
websites to visit) in the input dataset. Credit SAS Institute
at:
http://ftp.sas.com/techsup/download/sample/datastep/macnumob_sas.html
********/

%global dset nvars nobs;
 %let dset=&ds;
 %let dsid = %sysfunc(open(&dset));
 %if &dsid %then
   %do;
      %let nobs =%sysfunc(attrn(&dsid,nobs));  ****** nobs is macro
variable which keeps the number of observations in the dataset;
      %let rc = %sysfunc(close(&dsid));
   %end;
 %else
    %put open for data set &dset failed
         - %sysfunc(sysmsg());

/************* Code to determine the number of observations (i.e. -
websites to visit) in the input dataset ********/


%DO b=1 %TO &nobs;   ****** beginning of do loop through the
observations *******;
data SiteVisit;
      Do b=&nobs - &b;
      SET &ds point=b; ****** Grab observations one at a time *******;
      AccessionADD=COMPRESS("'http://www.sec.gov:80/Archives/"||Accessi
on||"'"); ****** Specify exact website location *******;
      CALL SYMPUT("SITE",AccessionADD); ****** Save website location as
a macro variable *******;
      output;
      end;
      stop;
      run;

data SiteVisit;

filename foo url &SITE debug; ****** Invokes url method - depending on
your internet access you may need to specify a port of proxy server
                                      See
http://www2.sas.com/proceedings/sugi28/073-28.pdf for more details
*******;

retain linecount line countA1 countB1 countC1 countD1 countE1 countA2
countB2 countC2 countD2 countE2 countE3 countF
file_date form_type name cik ein fyr accession smbl lagline gvkey;
length line $256 form_type $63 name $63 accession $66 cik $15 ein $15;

infile foo lrecl=256 pad expandtabs ; ****** Accesses specified web
location and reads line by line *******;
input line  $char256. ;

linecount=_n_;
```

```
If _n_ = 1 then do;  *** Variables which will be used to keep lines
surrounding our key words ****;
countA1=0 ;
countB1=0 ;
countC1=0 ;
countD1=0 ;
countE1=0 ;
countA2=0 ;
countB2=0 ;
countC2=0 ;
countD2=0 ;
countE2=0 ;
countE3=0 ;
countF=0 ;
end;


if line eq ' ' then return;
line1 = UPCASE(htmldecode(compress(line, ' ')));


******** Gather basic information about the SEC form: accession #, form
type, company name, cik, ein, fiscal year ****;
if index(line1,'ACCESSIONNUMBER:') ne 0 then do ;
      accession = tranwrd(line,'ACCESSION NUMBER:',' ') ;
      accession = trim(left(accession)) ;
end ;

if index(line1,'CONFORMEDSUBMISSIONTYPE:') ne 0 then do ;
      form_type = tranwrd(line,'CONFORMED SUBMISSION TYPE:',' ') ;
      form_type = trim(left(form_type)) ;
end ;

if index(line1,'FILEDASOFDATE:') ne 0 then do ;
      file_date = tranwrd(line,'FILED AS OF DATE:',' ') ;
      file_date = trim(left(file_date)) ;
end;

if index(line1,'COMPANYCONFORMEDNAME:') ne 0 then do ;
      name = tranwrd(line,'COMPANY CONFORMED NAME:',' ') ;
      name = trim(left(name)) ;
end ;

if index(line1,'CENTRALINDEXKEY:') ne 0 then do ;
      cik = tranwrd(line1,'CENTRALINDEXKEY:',' ') ;
      cik = trim(left(cik)) ;
end ;

if index(line1,'IRSNUMBER:') ne 0 then do ;
      ein = compress(tranwrd(line1,'IRSNUMBER:',' '),'][') ;
      ein = trim(left(ein)) ;
end ;

if index(line1,'FISCALYEAREND:') ne 0 then do ;
      fyr = tranwrd(line1,'FISCALYEAREND:',' ') ;
      fyr = trim(left(fyr)) ;
```

```
end ;

******** Gather basic information about the SEC form: accession #, form
type, company name, cik, ein, fiscal year ****;


lagline = lag1(line) ;   *** Save the last line ***;

*********** Turn on our counting variables when we see the key words
***********;
            if index(line1,'INCORPORATEDIN') ne 0 then do;
                  countA1 = countA1 + 1 ;
            end;
            if index(line1,'INCORPORATEDUNDER') ne 0 then do;
                  countA2 = countA2 + 1 ;
            end;
            if index(line1,'FOUNDEDIN') ne 0 then do;
                  countB1 = countB1 + 1 ;
            END;
            if index(line1,'FOUNDEDUNDER') ne 0 then do;
                  countB2 = countB2 + 1 ;
            END;
            if index(line1,'FORMEDIN') ne 0 then do;
                  countC1 = countC1 + 1 ;
            end;
            if index(line1,'FORMEDUNDER') ne 0 then do;
                  countC2 = countC2 + 1 ;
            end;
            if index(line1,'ORGANIZEDIN') ne 0 then do;
                  countD1 = countD1 + 1 ;
            end;
            if index(line1,'ORGANIZEDUNDER') ne 0 then do;
                  countD2 = countD2 + 1 ;
            end;
            if index(line1,'ENGAGEDIN') ne 0 then do;
                  countE1 = countE1 + 1 ;
            end;
            if index(line1,'SUCCEED') ne 0 then do;
                  countE2 = countE2 + 1 ;
            end;
            if index(line1,'SUCCESSORTO') ne 0 then do;
                  countE3 = countE3 + 1 ;
            end;
            if index(line1,'ITEM1') ne 0 then do;
                  countF = countF + 1 ;
            end;

*********** Turn on our counting variables when we see the key words
***********;

*********** Continue to count so that we can save several lines after
keywords ***********;
            if countA1 > 0 then countA1 = countA1 + 1 ;
            if countB1 > 0 then countB1 = countB1 + 1 ;
            if countC1 > 0 then countC1 = countC1 + 1 ;
            if countD1 > 0 then countD1 = countD1 + 1 ;
            if countE1 > 0 then countE1 = countE1 + 1 ;
            if countA2 > 0 then countA2 = countA2 + 1 ;
            if countB2 > 0 then countB2 = countB2 + 1 ;
            if countC2 > 0 then countC2 = countC2 + 1 ;
            if countD2 > 0 then countD2 = countD2 + 1 ;
            if countE2 > 0 then countE2 = countE2 + 1 ;
            if countE3 > 0 then countE3 = countE3 + 1 ;
            if countF > 0 then countF = countF + 1 ;
*********** Continue to count so that we can save several lines after
keywords ***********;

      run ;

data SiteVisit ; set SiteVisit ;

*********** Output lines surrounding keyword ***********;
if (0 < countA1 <= 5) then do;
output;
end;
if (0 < countB1 <= 5) then do;
output;
end;
if (0 < countC1 <= 5) then do;
output;
end;
if (0 < countD1 <= 5) then do;
output;
end;
if (0 < countE1 <= 5) then do;
output;
end;
if (0 < countA2 <= 5) then do;
output;
end;
if (0 < countB2 <= 5) then do;
output;
end;
if (0 < countC2 <= 5) then do;
output;
end;
if (0 < countD2 <= 5) then do;
output;
end;
if (0 < countE2 <= 5) then do;
output;
end;
if (0 < countE3 <= 5) then do;
output;
*********** Output lines surrounding keyword ***********;
end;


PROC Append Base=Cumulate data=SiteVisit; *********** Append results to
file in work library called "Cumulate" ***********;
Run;
%End;

%MEND webcrawler;
```

```
%webcrawler (ds=test_age);  ***** Input the location of the dataset
here - the output file will be called "Cumulate" and will be in the
work library **;
                        ***** The webcrawler assumes you have a dataset
that looks like the sample dataset with a variable called "accession"
that has the location of the SEC form - this variable is created after
running make_accession_masterlist.sas and filtering these observations
down to the forms of interest.  See the test_age.sas7bdat as an example
**;
Run;
```