

Use SAS[®] to Check Misspelled Words in Excel Files

Tse-Hua Shih, Ph. D., inVentiv Clinical Solutions, LLC., Baltimore, MD

Cliff Meng, Ph. D., inVentiv Clinical Solutions, LLC., Baltimore, MD

Eugenia Henry, Ph. D., inVentiv Clinical Solutions, LLC., Baltimore, MD

Yuan-Chi Chang, M.A., inVentiv Clinical Solutions, LLC., Baltimore, MD

Xitao Fan, Ph. D., University of Virginia, Charlottesville, VA

ABSTRACT

In the pharmaceutical industry, data are sometimes documented in excel files (e.g., define and mapping documents for 23 STDM domains, etc). To spot misspelled words in excel files, a straightforward method is to copy-and-paste contents of excel files onto word documents and do eye-ball checks pages after pages. It consumes lots of time and energy, which often results in human errors due to physical tiredness. This paper describes in details how SAS can check misspellings in excel files and instantly documents misspelled words, rows of data that contain misspellings, and variable names that have misspellings in variable values. The current paper presents a novel method to check spellings in excels files, which exponentially enhances efficiency and precision of such QC works and a dependable corporate image when presenting qualitative data.

MOTIVATION

If spelling checks in excel files can be done with one click of a SAS program before you leave your office, it means that work is done even when you stay at home. The current paper demonstrates how to realize this idea.

RATIONALE

In an article “Developing PDF-Manipulation Macros for eSubmission Automation” by Zhang (2006), there is SAS macro that collaborates with a VBScript program to retrieve misspelled words from a Portable Document Format (PDF) file. If excel data can be imported into SAS and then be exported into PDF files, it means that checking PDF files is the same as checking excel files. The current paper introduces how SAS can automate this process that basically comprises the following four steps.

Step 1: Import Each Sheet of Each Excel File

One can uses Pipe and Proc Import to import each sheet from each excel file as one SAS dataset, as shown below:

```
FileName MyDir Pipe "dir C:\Test /B /S";
DATA all_dir;
INFORMAT path $100.;
INFILE MyDir lrecl=300 trunccover;
INPUT @1 path ;
run;

data x;
  set all_dir;
  length file $100;
  if index (path, 'xls') > 0 ;
  file=scan(path,-1,'\');
  replacement='';
  file1 = "\"||Trim(left(file));
  add=tranwrd(path,trim(file1),replacement);
  put add;
  name=scan(file,1,'. ');
  proc sort;
  by add name;
  run;

data _null_;
  set x;
  call symput ("add"||trim(left(_n_)),trim(left(add)) );
  call symput ("file"||trim(left(_n_)),trim(left(file)) );
  call symput ("name"||trim(left(_n_)),trim(left(name)) );
```

```

call symput('import',trim(left(_n_)));
run;

%macro get_sheet (start);
%do x=&start %to &import;
LIBNAME exbk excel "&&add&x\&&file&x" ;

PROC SQL;
    CREATE TABLE s_&&name&x AS
    SELECT distinct memname
    FROM DICTIONARY.COLUMNS
    WHERE libname='EXBK' AND memtype='DATA' and index(memname,'$');
QUIT;

LIBNAME exbk clear;

DATA s_&&name&x; SET s_&&name&x;
memname = compress(memname, "$"); * remove the $ and quotation marks ;
add = "&&add&x";
name = "&&name&x";
proc sort;
by add name;
run;
%end;

%mend;
%get_sheet(1);

%macro merge(start);
data import_list;
merge x %do x=&start %to &import; s_&&name&x %end; ;
by add name;
run;
%mend;
%merge(1);

data _null_;
    set import_list;
call symput ("add"||trim(left(_n_)),trim(left(add)) );
call symput ("file"||trim(left(_n_)),trim(left(file)) );
call symput ("val"||trim(left(_n_)),trim(left(memname)) );
call symput ("name"||trim(left(_n_)),trim(left(name)) );
call symput('xx',trim(left(_n_)));
run;

%macro import(start);
%do j=&start %to &xx;
PROC IMPORT DATAFILE= "&&add&j\&&file&j"
    OUT = &&name&j.._&&val&j(keep= _all_) DBMS=excel2000 REPLACE;
GETNAMES=yes;
SHEET="&&val&j";
RUN;
%end;
%mend;
%import (1);

```

Step 2: Export to PDF Files

One first uses ODS PDF to export each SAS dataset into a PDF file, as shown

below:

```
options ORIENTATION = LANDSCAPE NOCENTER nodate nonumber;
ods escapechar="^";
ods pdf file="H:\Test\&&val&k...pdf";
proc print data=&lib..&&val&k ;
run;
ods pdf close;
```

Step 3: Spelling Checks

The article of Zhang (2006) drafted the code below to retrieve misspellings from exported PDF files. Please obtain the VBScript program from Zhang's article.

```
filename _SCRIPT_ PIPE
"cscript H:\Test\S.vbs //NOLOGO H:\Test\&&val&k...pdf"
console=min;

data a;
length ID Word Suggestion $256;
infile _SCRIPT_ missover;
input;
ID=input(scan(_infile_,1,:), best12.);
Word=scan(_infile_,2);
Suggestion=scan(_infile_,3);
keep ID Word Suggestion;
run;
```

Step 4: Post-Processing

One can treat variable names and variable formats in each SAS dataset as macro variables. After converting all variables into character variables, a user can use INDEX function to see if any value of a variable contains any retrieved misspelling from dataset "a" in the 3rd step. In this way, we can check if there is any misspelling in variable

formats. Of course, if a user keeps only character variable before running the macro, codes in the current step can be simplified.

```
%let dsid=%sysfunc(open(&lib..&&val&k));
%let cnt=%sysfunc(attrn(&dsid,nvars));
%do i = 1 %to &cnt;
%let x&i=%sysfunc(varname(&dsid,&i));
%end;

%let rc=%sysfunc(close(&dsid));

data d_&&val&k;
    set &lib..&&val&k;
%do i=1 %to &cnt;
call symput ("f&i", vformat(&&x&i));
%end;
run;

data d1_&&val&k;
    set d_&&val&k;
%do i=1 %to &cnt;
c_&&x&i = put(&&x&i, &&f&i);
%end;
drop %do i=1 %to &cnt; &&x&i %end;;
run;

data xxx.c_&&val&k;
length word $ 50 variable $ 50;
set d1_&&val&k;
%do i=1 %to &cnt;
%do j=1 %to &t;
if vtype(c_&&x&i)='C' and index (c_&&x&i,"&&v&j") > 0 then do;
word = "&&v&j"; variable = "&&x&i"; output;
end;
/*document misspelled words, rows of data that contain misspellings,
and variable names that have misspellings in format or variable
values.*/
%end;
%end;
run;
```

LIMITATION

Currently, this approach of spelling checks, through VBScript, may take up some CPU time especially when a excel file is big. After the program finishes running, one still needs use his/her own discretion to choose among retrieved misspellings, because some

(e.g., variable names, abbreviations) can not be recognized by the VBScript as legitimate spellings.

CONCLUSION

To check misspellings in excel files, an employer can choose to slave numerous eyeballs or even purchase some special software. The current paper demonstrates a practical wisdom to let SAS do this job for you. While you refresh yourself and spend quality time with family, you know well that your job is also done by SAS. What can be merrier than this! To overcome limitations of this study, readers of the current article can apply the idea of this paper to use an undocumented SAS function (i.e., Proc Spell) for the same job, just faster.

REFERENCES

Zhang, L. (2008). Developing PDF-Manipulation Macros for eSubmission Automation. *Paper Presented at the Annual Conference of PharmaSUG at Atlanta, Georgia.* <http://www.lexjansen.com/pharmasug/2006/applicationsdevelopment/ad01.pdf>

CONTACT INFORMATION

Your comments and questions can be directed to:

Shih Tse-Hua (Jeremy)

Email: jshih@inventivclinical.com

TRADEMARK INFORMATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.