Paper 287-2011

Visualizing Count Data with a 3-D Flash Animation in SAS® Environment

Xin Wei, Independent Consultant, Morris Plains, NJ

ABSTRACT

Modern data visualization technology requires the integration of multiple tools and platforms. This paper describes using Base SAS® to wrap a JavaScript, a Flash object, and XML in HTML text in order to visualize the word count of the pharmaceutical SAS user conference proceedings from 2000 to 2010 in three-dimensional animation.

INTRODUCTION

In tradition statistical graphics, a statistical quantity is proportionally visualized by a static plotting entity such as bar height of bar chart and arc length of pie chart. Bubble plot utilizes the size or area of a plotting unit to demonstrate the third dimension of data points. We recently see a type of plot called “tag cloud” in blog sphere for the visualization of the most frequently used words in web posting, in which the size of a textual token represents the frequency of the token appearing in a collection of texts. Figure 1 shows a tag cloud from http://www.lexjansen.com/sugi/, demonstrating the popular words used by SUGI papers. To bring more flavors into the world of static graphics, Roy Tanck in his blog proposes to use a flash movie to visualize the word frequency in a 3-D animation (http://www.roytanck.com/2008/05/19/how-to-repurpose-my-tag-cloud-flash-movie/) [1]. The basic idea behind this is that the plotting properties of word count is encoded in XML format and passed to a flash object by a JavaScript. In this paper, we demonstrate how to use base SAS to do word count, create XML and call flash object and JavaScript in a pure text. This process, together with flash object and JavaScript that are readily available to download, produce a word tag cloud for PharmaSUG in a web browser with a refreshing 3-D effect (http://www.xinwei2010.com/sastagcloud/pharmasug.html, press ctrl and F5 together if the page fails to load, you need Flash Player 9 to play this animation.)

Generate Word Count

For the demonstration purpose, we aim at producing a 3-D tag cloud that summarizes the word count for the presentations from Pharmaceutical SAS User Group in the last 10 years. We use filename URL statement in base SAS to download the contents from http://www.lexjansen.com/pharmasug/ which hosts collections for papers/presentations from various SAS User Groups. Although not trivial, the implementation of this approach has been discussed with great details in our 2010 PharmaSUG paper [2]. Therefore we won’t repeat every technical aspects of this but rather provide a general outline. Interested audience may read our paper to get an in-depth understanding of this process.

1. Read the html pages for PharmaSUG 2000-2010 using URL method

```
%do i=0 %to 10;
%if &i<10 %then %let i=%sysfunc(putn(&i,z2.));
filename fetch url "http://www.lexjansen.com/cgi-
bin/xsl_transform.php?x=psug&i%nrstr(&s=pharmasug&c=pharmasug)"
debug lrecl=8192;

data pgcnt;
     infile fetch length=len;
     input record $varying8192. len;
run;
*****code not shown******;
%end;
```

The above macro code loops through all 10 html pages for the last ten years PharmaSUG proceedings and aggregate all text contents from web site into a SAS dataset pgcnt. Unfortunately, the full text is only available in PDF format that cannot be text-mined by base SAS. As a result, we can only do word count on the titles of presentation that are in the text html format.

2. Parse out presentation title from html body

```
data title;
     set pgcnt;
     length title temp $8192;
     format title temp $varying8192.;
     if index(record,'href="http://www.lexjansen.com/pharmasug') and
     index(record,'.pdf');
     temp=tranwrd(record,'.pdf">','~');
     title=scan(temp,2,'~');
     title=scan(title,1,'<');
run;
```

The presentation title strings are always preceded by the PDF paper URL address therefore can be located by this characteristic using index( ) function. Then they are parsed out by scan( ) function based on their positions in html syntax relative to some special characters.

3. word count

```
data list;
     set title;
     length word $30;
     i=1;
     do while(scan(title,i,' ') ne '');
         word=scan(title,i,' ');
         if substr(upcase(word),1,3)='SAS' then word='SAS';
         i+1;
         output;
     end;
     keep word title year;
run;
proc freq data=list noprint;
     table word/out=count;
run;
```

Each individual word is extracted from the title line by scan( ) function with blank as delimiter. The word count is then calculated by **proc freq** from this long single word list. However, the resulting word count isn't that informative because it contains many common English words that do not particularly concern the topic of our interest. We download "the 100 most commonly used English words" from web and read them into SAS. The popular words used by SAS presentation are deleted if they are also the commonly used English word. The codes that achieve above are shown as follows:

```
proc import datafile='U:\work\sas proceedings\SGF2011\common_words.xls'
     out=common replace;
run;
proc sql noprint;
     create table cnt as
     select *, 'http://www.lexjansen.com/pharmasug/' as url from count
     where word not in ('-') and
     lowcase(trim(left(word))) not in (select lowcase(trim(left(word)))
     from common) order by count desc;
quit;
```

Turn the tag count into a 3-D flash animation

Upon preparing this manuscript, we with delight discover a SESUG paper that demonstrates the creation of a static tag cloud with SAS ODS [3]. Figure 1 shows another impressive static tag cloud from http://www.lexjansen.com/sugi/ that showing the word frequency of SUGI papers. Here we move one step further to visualize a 3-D tag cloud in a flash movie by integrating a flash object, JavaScript and XML.

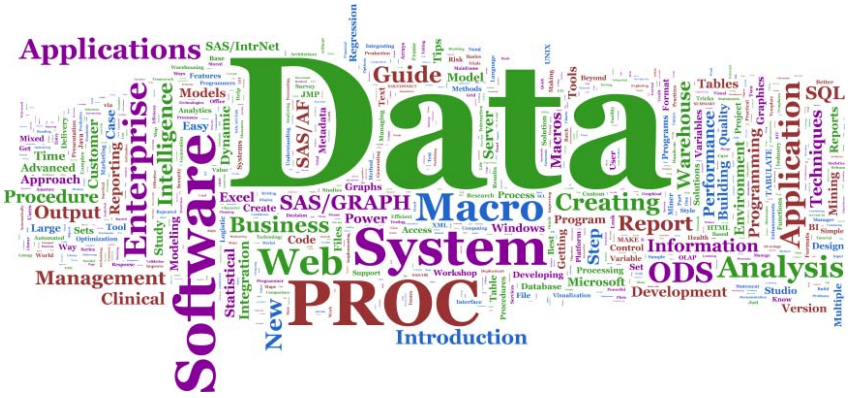

Figure 1

1. Framework layout in a html body file

```
data html;
     infile cards4 truncover;
     input line $1-30000;
cards4;
<head>
       <title>PharmaSUG tagcloud</title>
       <meta http-equiv="Content-Type" content="text/html" />
    #####codes not shown######
</body>

</html>
;;;;
run;
```

First of all, cards4 statement is called to input complex html syntax with a number of special characters including semicolon. The rest of codes in this section are the syntax that needs to be entered in the data step.

```
<head>
    <title>PharmaSUG tagcloud</title>
    <meta http-equiv="Content-Type" content="text/html" />
    <!-- SWFObject embed by Geoff Stearns geoff@deconcept.com http://blog.deconcept.com/swfobject/ -->
    <script type="text/javascript" src="swfobject.js"></script>
    <style type="text/css">
        body { background-color: #eee; padding: 20px; }
    </style>
</head>
```

Calling a JavaScript to render tag information to flash object

The boxed part indicates that a JavaScript “swfobject.js” needs to be loaded to render the tag information as follows to a flash object.

The following syntax is a piece of JavaScript that is going to be embedded in the html page.

```
<body>

    <div id="flashcontent">This will be shown to users with no Flash or Javascript.</div>

    <script type="text/javascript">
        var so = new SWFObject("tagcloud.swf", "tagcloud", "1000", "800", "7", "#ffffff");
        // uncomment next line to enable transparency
        //so.addParam("wmode", "transparent");
        so.addVariable("tcolor", "0x333333");
        so.addVariable("mode", "tags");
        so.addVariable("distr", "true");
        so.addVariable("tspeed", "100");
        so.addVariable("tagcloud", "#");
        so.write("flashcontent");
    </script>

</body>

</html>
```

Dimension of movie in browser

Embedding a flash object

Tag XML data need to be filled in

Animation setting

What it does is to render a tag cloud of the list of popular words in a flash object “tagcloud.swf” in a transparent background at 1000 by 800 pixels. The parameters set by “so.addVariable” define the default setting of animation such as the speed of rotation, token distribution on the rolling sphere and background color. The last “so.addVariable” statement is supposed to define the flash variable “tagcloud” with a XML data for the flash movie. Here we put “#” in code as place holder that will be replaced by XML code produced by SAS in the next step.

2. Generate XML data that define the characteristics of animation

The flash variable “tagcloud” anticipates a XML data as follows:

```
<tags>
    <a href='http://www.lexjansen.com/pharmasug/' style='15'
      color='0x006600' hicolor='0x00FF00'>SAS</a>
    <a href='http://www.lexjansen.com/pharmasug/' style='13'
      color='0xCC3399' hicolor='0xFFCCFF'>Data</a>
</tags>
```

As we may see, each line defines the words (for instance, “SAS”,”Data”) that appear in the animation and their size, URL, default color and mouse over color. The XML code for our example will be created from the word count data “cnt” we generated before. The count or frequency of words can easily be converted to token font. Hex color code can be found in the following link: http://www.nthelp.com/colorcodes.htm. Once read into SAS, the color codes are merged to count data so that each word is randomly assigned to different default color and hover color after the sequence of color codes are re-shuffled by being sorted with a random number.

```
proc import datafile='U:\work\sas proceedings\SGF2011\color_code.xls'
     out=color_code replace;
run;
data color_code;
     set color_code;
     order=rannor(1);
run;
proc sort data=color_code out=color(rename=(color_code=color));
     by order;
run;
proc sort data=color_code out=hicolor(rename=(color_code=hicolor));
     by descending order;
run;
data cntcolor;
     merge cnt color hicolor;
run;
```

The following SAS codes concatenate the word, and its token size, URL, color and hover color into one column “info”

in SAS dataset. Then the resulting strings for each of the top 20 or so words are further concatenated into a big chunk of XML text stored in a SAS macro variable “infolist”. Here we adjust the token size by logarithm transformation on count frequency so that the word font would be compatible to the size of animation window. The macro variable &num_word make easy for end user to control the how many words appear in the animation.

```
data info;
     set cntcolor;
     length info $1000;
     *count=int(log2(count))+5;
     count=LOG10(count)/LOG10(1.6);
     *count=count/10;
     if _n_=1 then  info="<tags>"; output;
     info="<a href='http://www.lexjansen.com/pharmasug/' style="
          ||"'"||trim(left(put(count,best.)))||"' "||"color="||"'"
          ||trim(left(color))||"' hicolor="||"'"||trim(left(hicolor))
          ||"'"||'>'||trim(left(word))||"</a>";
     output;
     if _n_=&num_word then do;
        info="</tags>";
        output;
       stop;
     end;
run;

proc sql noprint;
     select trim(left(info)) into: infolist separated by '' from info;
quit;
```

3. Embed the XML data into the body HTML file

The final step is to combine the products of the previous two steps in this section. First of all, the XML data is inserted into the statement that define flash variable “tagcloud” via SAS macro variable by tranwrd( ) function. So far, a functional html text has been produced in SAS dataset. Once this SAS dataset is converted to a pure text html, a vivid 3-D animation of rotating globe filled by flying words can be open from it in a web browser by a SAS dm statement at the end of code execution (shown in http://www.xinwei2010.com/sastagcloud/pharmasug.html , press ctrl and F5 together if the page fails to load, you need a Flash Player 9 to be able to play this animation.). The screen shot of this animation is shown in Figure 2. Unlike a crowed static tag cloud, even words with the smallest font are visible in it because every word is brought to the front of viewer during the animation. The speed and orientation of rotation can be control by the location of cursor. We can click the word in the rotating cloud to open a PharmaSUG proceeding page because each word can be assigned to a distinct URL in XML.

```
data html;
     set html;
     if index(line,'so.addVariable("tagcloud", "#");')
     then line=tranwrd(line,'#',"&infolist");
run;
data _null_;
     file 'U:\work\sas proceedings\SGF2011\wp-cumulus-
           example\pharmasug.html' lrecl=30000;
     set html;
     put line;
run;
dm "wbrowse 'U:\work\sas proceedings\SGF2011\wp-cumulus-
    example\pharmasug.html'";
```

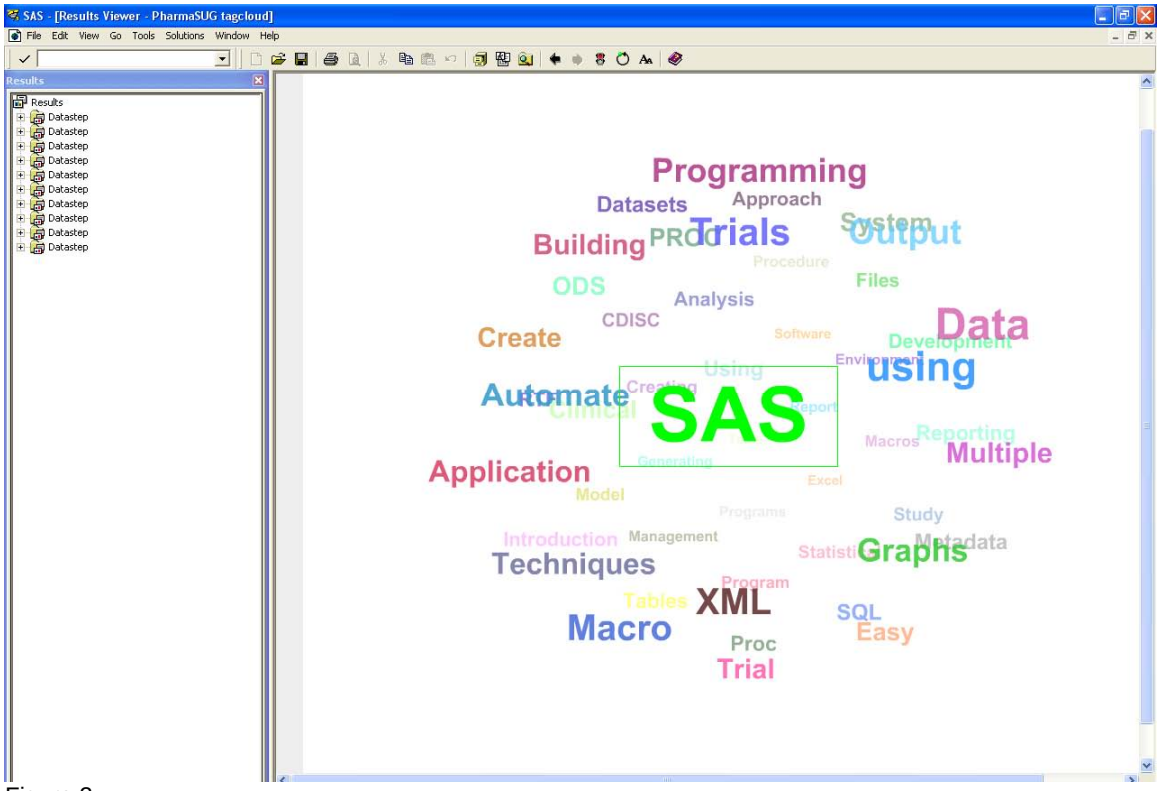

Figure 2

4. Requirement

Although here SAS plays a pivotal role in streamlining the process, we have to admit that flash object “tagcloud.swf” and JavaScript “swfobject.js” have done most of the heavy lifting stuffs. They are freely available for download in Roy Tanck’s web site http://www.roytanck.com/2008/05/19/how-to-repurpose-my-tag-cloud-flash-movie/ . They need to sit in the same folder as your main html file on your local drive or web server. The direct link for download is http://www.roytanck.com/wp-content/uploads/wp-cumulus-example.zip. The zip file contains swf and js files and a sample html page. You need at least Flash Player 9 to visualize this animation.

REFERENCES

1. http://www.roytanck.com/2008/05/19/how-to-repurpose-my-tag-cloud-flash-movie/
2. Use Base SAS URL to Build Surveillance and Monitoring System for New Clinical Trial Registration. Xin Wei, James Cai, Jim Rosinski, Hoffmann-la-Roche. PharmaSUG 2010 Proceedings. http://www.pharmasug.org/cd/papers/AD/AD23.pdf
3. Tag Clouds - A list of tokens, sized by relative frequency. Richard A. DeVenezia. SESUG Proceedings 2008. http://analytics.ncsu.edu/sesug/2008/SIB-096.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Xin Wei
Enterprise: Independent Consultant
Address:
City, State ZIP: Morris Plains, NJ, 07950
Work Phone: 973-722-7139
Fax:
E-email: xinwei@stat.psu.edu
Web: