Using Correspondence Analysis to Analyze Canadian Federal Budget Speeches, 1966 – 2013

Brian Greiner
brian@greiner.ca

April 21, 2013

SUMMARY

Correspondence analysis is used to examine Canadian federal budget speeches to see what, if any, patterns exist in the stated budgetary intentions of the ruling political parties. The analysis shows how the tone of the budget speech can change over time, even for the same political party, as well as highlighting differences between the speeches given by different political parties. The analysis can also be indicative of major political events, such as change in leadership or elections. The analysis was performed using programs written in the R language, and making use of existing libraries.

INTRODUCTION

In the political arena, budget speeches are used by the government to declare the nominal spending patterns of their focus for the next year or so. Examining these speeches for clues to the government's intentions is typically likened to the ancient practice of reading entrails. And, of course, there are always vast differences between stated intentions and actual practice. Still, it would be interesting to see what, if any, patterns could be discovered in and between these speeches. At the very least it would be instructive to see if these speeches reflect any differences between political parties.

A popular way to display the relative frequency of words in a document is by the use of a “word cloud”, which displays the words with the size of the word related to its frequency. This was, in fact, my first approach to looking at budget speeches. While it made a pretty picture, it was unsatisfying due to its non-quantitative nature. However, for those who enjoy such things, Macleans magazine has published a series of word clouds of the Federal budget speeches from 1995 -2013.

While investigating more quantitative approaches to analyzing text documents, I came across the concept of “correspondence analysis”. According to Wikipedia, “Correspondence analysis (CA) is a multivariate statistical technique proposed by Hirschfeld[2] and later developed by Jean-Paul Benzécri. It is conceptually similar to principal component analysis, but applies to categorical rather than continuous data. In a similar manner to principal component analysis, it provides a means of displaying or summarizing a set of data in two-dimensional graphical form.”

Sinclair discusses how correspondence analysis can be used to look for patterns in sets of textual data. Bendixen discusses correspondence analysis as a technique for reducing multidimensional data into a simpler

form that can then be more easily visualized for analysis.

DATA

A total of 43 speeches were acquired, ranging from 1966 – 2013 (excluding 1967). Speeches for the years 1968 to the present are available either in a PDF or on a separate web page. The years prior to 1968 are only available as part of Hansard, the official record of parliamentary debate. The Hansard copies appear to be images of the original documents, which can make converting them to a text file tedious and problematic. For those reasons, only the speech for 1966 was processed from the Hansard copy. An attempt to process the speech for 1967 was made, but was abandoned when it became apparent that line-by-line extraction would be required.

The speeches were obtained by scraping web pages and from PDF documents, and saved as text files. Some of the resulting text files contained extended ASCII characters that confused the analysis programs, so I created a program (PROGRAM-3 below) to replace the specific problem characters. These files, and all the analysis programs, are available on request.

TABLE-1 shows the dates of the speeches used, along with the names of the Prime Minister and Finance Minister that presented the budget. The years when an election was held is also indicated, since budgets for those years are typically presented just prior to the calling of an election.

DATA ANALYSIS

The files to be analyzed were read in as a corpus using the “tm” package. The corpus of documents was analyzed using correspondence analysis with the “FactoMineR” package.

The output of the correspondence analysis is a set of multidimensional eigenvalues, with each dimension contributing to understanding the associations that are present in the data. The results of the analysis are shown in TABLE-2. The amount that each dimension contributes is shown in the “percentage of variance” column. This is shown graphically in FIGURE-1. These show that the first two dimensions account for the largest incremental explanation of the corpus data, with subsequent dimensions contributing smaller incremental amounts.

The dimension data can be used as the basis of a distance calculation, which in turn can be used to determine the clustering between speeches. This clustering, calculated using the hclust() function, can be expressed graphically as a cluster denogram. The distance was calculated using a “euclidean” measure, which is simply the square distance between the two distance vectors. After experimenting with clustering methods, I settled on the “centroid” method as it gave the most visually pleasing results. It should be noted, however, that the choice of method had little effect on the calculated clusters.

The results of the analysis can be sorted to show the most frequently used words. TABLE-3 shows the top 20 words for each of the speeches, sorted by party and year.

DISCUSSION

One of the biggest attractions of Correspondence Analysis is its ability to reduce data to a minimum number of dimensions, and produce useful x-y plots. FIGURE-2 shows a plot based on the first two dimensions of the analysis.

There are a number of interesting features about this graph. The most obvious feature is how closely the different speeches lie on a U-shaped curve. That is, the tone of the speeches seems to change over time, typically in a smooth fashion. It would appear that the tone of the speeches seems to continuously and smoothly evolve over time. If this is the case, then it would be instructive to look at the flow from year to year. Closer examination shows that there are several instances where this smooth flow can jerk or jump suddenly in a different direction. It would also be interesting to see if any obvious cause for this can be determined.

One large jump occurs between 1970 and 1971. Looking at TABLE-1, it can be seen that although the principal players remain the same in these years, in 1972 a new Finance Minister entered the scene. One could hypothesize that, given the large change in tone of the budget speech, that the “new thinking” indicated by the change is somehow linked to the new Finance Minister.

The sideways jerk in 1972 coincides with an election. Interestingly, in 1973 the tone of the budget speech appears to swing back to the expected curve.

The 1974 speech makes another large jump, but that is an election year. After that election, things seem to swing back to the expected curve.

Interestingly, the election of a new party in 1979 doesn't seem to shift the dimensions of the budget speech away from the curve. As the song says “meet the new boss, same as the old boss”.

The speeches tend to “follow the curve” until 1985-1987, when a new party comes into power. These three years aren't too far off the curve, but interestingly seem to cluster together.

The 1992 speech moves away from the curve, and it precedes an election.

Another interesting feature is shown in the change of the speeches between the Progressive Conservative and Conservative governments. The difference between the two is that the PC's were essentially eliminated in the election of 1993, then were effectively taken over by the Reform Party, and renamed the “Conservative Party”. Once the Conservative Party gained power in 2006, the dimensions of its budget speeches had a lot more variance than had heretofore been seen. See, for example, the large change between the 2007 and 2008 budget speeches (2008 was an election year). Interestingly, the 2008 speech lies on the “normal” U-curve defined by the Liberals, if somewhat ahead of the curve. Interestingly, the trajectory of the Conservative speeches seems to form a rough ellipse, circling around rather than moving in a smooth curve as had heretofore been the norm. Another interesting feature is that the years 2007 and 2012 are almost identical in their dimension measurements, but I cannot think of any obvious explanation for this.

It is important, however, not to read too much into the graph. The dimensions of the analysis, for example, bear no relationship whatsoever to the political conveniences of “left” and “right”. Another way of looking at the similarities between speeches is to use the denogram clustering technique, illustrated in FIGURE-3.

The cluster analysis groups speeches by a calculated distance, based on all their dimensional data – as compared to the graph in FIGURE-2 which uses only the first two dimensions. These clusters show how the Conservative speeches form their own cluster, while the PC speeches are intermixed with the Liberal speeches of the time. Speeches that occur at special times, such as elections, clearly stand out as different from the rest.

CONCLUSIONS

Overall, the budget speeches appear to change smoothly over time. That is, change appears to be evolutionary rather than discontinuous. There is, however, the occasional jump and jerk. The analyzed dimensions of budget speeches tend to change prior to elections. This is not surprising, since governments tend to tweak their pre-election budgets to make themselves seem more attractive. Correspondence analysis confirms this, and gives us a measure of how much a government's message changes in election years.

The tone of the speeches can also be shown to vary depending on the Prime Minister and Finance Minister. This is not surprising, since these individuals have a large influence on the budget speech.

Although analyzing budget speeches can be instructive, it is important to keep in mind that budget speeches are statements of intention, which will bear varying degrees of correspondence to the government's actions.

DIRECTIONS FOR FUTURE RESEARCH

In addition to the budget speeches, the government gives a “speech from the throne” which is meant as a thumbnail sketch of the direction they intend to take for the current session of Parliament. It would be instructive to analyse these speeches using these same techniques. It might also be interesting to somehow correlate these two types of speeches for a given year, but at this point I'm not sure how that could be done. One approach might be to simply plot both sets of data as independent curves on a common graph. Preliminary work has begun on acquiring throne speeches.

Another instructional direction would be to somehow correlate the promises of the budget speech with what was actually done. At this point I'm not really sure how this could be accomplished by anything other than a forensic audit.

It would also be interesting to apply these techniques to the budget speeches on a provincial level. Work has begun on the budget speeches for Ontario, and this will be reported on in the near future (hopefully).

REFERENCES

Bendixen, Mike — A Practical Guide to the Use of Correspondence Analysis in Marketing Research
http://marketing-bulletin.massey.ac.nz/V14/MB_V14_T2_Bendixen.pdf

Budget Speeches — http://www.budget.gc.ca/pdfarch/index-eng.html

FactoMineR — Multivariate Exploratory Data Analysis and Data Mining with R
http://cran.r-project.org/web/packages/FactoMineR/index.html

Heuristic Andrew — Text Data Mining with Twitter and R
http://heuristically.wordpress.com/2011/04/08/text-data-mining-twitter-r/

hclust() — http://stat.ethz.ch/R-manual/R-patched/library/stats/html/hclust.html

Kabacoff, Rob — Quick-R : Cluster Analysis
http://www.statmethods.net/advstats/cluster.html

Macleans Magazine — Picturing Budget Speeches
http://www2.macleans.ca/2012/03/29/picturing-budget-speeches/

Sanchez, Gaston — Mining Twitter With R
http://sites.google.com/site/miningtwitter/questions/talking-about/given-users

Sinclair, Stefan — A Gentle Introduction to Correspondence Analysis
http://stefansincalir.name/correspondence-analysis

tm — Text Mining Package
http://cran.r-project.org/web/packages/tm/index.html

Wikipedia — Correspondence Analysis
http://en.wikipedia.org/wiki/Correspondence_analysis

1968	Lib	Pierre Trudeau	Edgar Benson	ELECTION
1969	Lib	Pierre Trudeau	Edgar Benson	
1970	Lib	Pierre Trudeau	Edgar Benson	
1971	Lib	Pierre Trudeau	Edgar Benson	
1972	Lib	Pierre Trudeau	John Turner	ELECTION
1973	Lib	Pierre Trudeau	John Turner	
1974	Lib	Pierre Trudeau	John Turner	ELECTION
1975	Lib	Pierre Trudeau	John Turner	
1976	Lib	Pierre Trudeau	Donald Macdonald	
1977	Lib	Pierre Trudeau	Donald Macdonald	
1979	PC	Joe Clark	John Crosbie	ELECTION
1980				ELECTION
1981	Lib	Pierre Trudeau	Allan MacEachen	
1983	Lib	Pierre Trudeau	Marc Lalonde	
1984	Lib	Pierre Trudeau	Marc Lalonde	ELECTION
1985	PC	Brian Mulroney	Michael Wilson	
1986	PC	Brian Mulroney	Michael Wilson	
1987	PC	Brian Mulroney	Michael Wilson	
1988	PC	Brian Mulroney	Michael Wilson	ELECTION
1989	PC	Brian Mulroney	Michael Wilson	
1990	PC	Brian Mulroney	Michael Wilson	
1991	PC	Brian Mulroney	Michael Wilson	
1992	PC	Brian Mulroney	Don Mazankowski	
1993				ELECTION
1994	Lib	Jean Cretien	Paul Martin	
1995	Lib	Jean Cretien	Paul Martin	
1996	Lib	Jean Cretien	Paul Martin	
1997	Lib	Jean Cretien	Paul Martin	ELECTION
1998	Lib	Jean Cretien	Paul Martin	
1999	Lib	Jean Cretien	Paul Martin	
2000	Lib	Jean Cretien	Paul Martin	ELECTION
2001	Lib	Jean Cretien	Paul Martin	
2003	Lib	Jean Cretien	John Manley	
2004	Lib	Jean Cretien	Ralph Goodale	ELECTION
2005	Lib	Jean Cretien	Ralph Goodale	
2006	CON	Stephen Harper	Jim Flaherty	ELECTION
2007	CON	Stephen Harper	Jim Flaherty	
2008	CON	Stephen Harper	Jim Flaherty	ELECTION
2009	CON	Stephen Harper	Jim Flaherty	
2010	CON	Stephen Harper	Jim Flaherty	
2011	CON	Stephen Harper	Jim Flaherty	ELECTION
2012	CON	Stephen Harper	Jim Flaherty	

TABLE-1 : budget year – Prime Minister – Finance Minister

	eigenvalue	percentage of variance	cumulative percentage of variance
dim 1	2.859e-01	6.744e+00	6.744
dim 2	1.855e-01	4.375e+00	11.119
dim 3	1.570e-01	3.704e+00	14.823
dim 4	1.512e-01	3.566e+00	18.389
dim 5	1.460e-01	3.443e+00	21.831
dim 6	1.374e-01	3.242e+00	25.073
dim 7	1.343e-01	3.167e+00	28.240
dim 8	1.259e-01	2.970e+00	31.210
dim 9	1.240e-01	2.925e+00	34.135
dim 10	1.196e-01	2.822e+00	36.957
dim 11	1.148e-01	2.707e+00	39.664
dim 12	1.121e-01	2.644e+00	42.308
dim 13	1.112e-01	2.623e+00	44.930
dim 14	1.078e-01	2.544e+00	47.474
dim 15	1.068e-01	2.520e+00	49.994
dim 16	1.047e-01	2.468e+00	52.463
dim 17	1.018e-01	2.401e+00	54.863
dim 18	1.005e-01	2.370e+00	57.233
dim 19	9.850e-02	2.323e+00	59.556
dim 20	9.797e-02	2.311e+00	61.867
dim 21	9.679e-02	2.283e+00	64.150
dim 22	9.458e-02	2.231e+00	66.381
dim 23	9.213e-02	2.173e+00	68.554
dim 24	9.090e-02	2.144e+00	70.698
dim 25	8.812e-02	2.078e+00	72.776
dim 26	8.680e-02	2.047e+00	74.823
dim 27	8.415e-02	1.985e+00	76.808
dim 28	8.217e-02	1.938e+00	78.746
dim 29	8.089e-02	1.908e+00	80.654
dim 30	7.994e-02	1.886e+00	82.540
dim 31	7.499e-02	1.769e+00	84.309
dim 32	7.418e-02	1.750e+00	86.058
dim 33	7.388e-02	1.743e+00	87.801
dim 34	7.316e-02	1.725e+00	89.526
dim 35	7.101e-02	1.675e+00	91.201
dim 36	7.029e-02	1.658e+00	92.859
dim 37	6.816e-02	1.608e+00	94.467
dim 38	6.539e-02	1.542e+00	96.009
dim 39	6.334e-02	1.494e+00	97.503
dim 40	5.436e-02	1.282e+00	98.785
dim 41	5.085e-02	1.199e+00	99.984
dim 42	6.687e-04	1.577e-02	100.000
dim 43	9.453e-33	2.230e-31	100.000

Table-2 : Details of analysis dimensions

	Con_1979.txt	Con_1985.txt	Con_1986.txt	Con_1987.txt	Con_1988.txt	Con_1989.txt	Con_2006.txt
[1,]	"tax"	"government"	"tax"	"tax"	"tax"	"tax"	"canadians"
[2,]	"energy"	"tax"	"government"	"economic"	"economic"	"government"	"government"
[3,]	"government"	"canadians"	"cent"	"government"	"government"	"canada"	"speaker"
[4,]	"cent"	"economic"	"canada"	"canada"	"growth"	"debt"	"tax"
[5,]	"canada"	"growth"	"billion"	"canadians"	"economy"	"billion"	"budget"
[6,]	"oil"	"budget"	"spending"	"economy"	"world"	"cent"	"canadian"
[7,]	"prices"	"canada"	"budget"	"deficit"	"canada"	"measures"	"canada"
[8,]	"price"	"increase"	"growth"	"fiscal"	"canadians"	"economic"	"million"
[9,]	"billion"	"billion"	"canadian"	"cent"	"canadian"	"income"	"support"
[10,]	"canadian"	"measures"	"canadians"	"growth"	"fiscal"	"sales"	"federal"
[11,]	"1980"	"jobs"	"program"	"program"	"jobs"	"budget"	"help"
[12,]	"budget"	"actions"	"measures"	"progress"	"trade"	"canadians"	"billion"
[13,]	"deficit"	"debt"	"business"	"billion"	"deficit"	"program"	"care"
[14,]	"economic"	"cent"	"fiscal"	"system"	"income"	"continue"	"families"
[15,]	"increases"	"deficit"	"income"	"continue"	"system"	"federal"	"cent"
[16,]	"measures"	"effective"	"jobs"	"income"	"progress"	"million"	"effective"
[17,]	"growth"	"sector"	"programs"	"reform"	"spending"	"rates"	"child"
[18,]	"increase"	"private"	"system"	"rates"	"agreement"	"fiscal"	"country"
[19,]	"capital"	"system"	"deficit"	"198788"	"business"	"programs"	"income"
[20,]	"federal"	"federal"	"financial"	"sales"	"cent"	"spending"	"providing"

	Con_2007.txt	Con_2008.txt	Con_2009.txt	Con_2010.txt	Con_2011.txt	Con_2012.txt	Con_2013.txt
[1,]	"canada"	"canada"	"canada"	"canada"	"canada"	"canada"	"canada"
[2,]	"canadians"	"speaker"	"action"	"government"	"canadian"	"canadians"	"canadians"
[3,]	"jobs"	"budget"	"economic"	"jobs"	"government"	"jobs"	"government"
[4,]	"government"	"tax"	"canadian"	"canadians"	"global"	"government"	"canadian"
[5,]	"term"	"billion"	"canadians"	"budget"	"jobs"	"term"	"job"
[6,]	"economic"	"people"	"plan"	"canadian"	"economy"	"economic"	"plan"
[7,]	"country"	"canadians"	"help"	"growth"	"economic"	"plan"	"jobs"
[8,]	"plan"	"million"	"government"	"businesses"	"canadians"	"country"	"growth"
[9,]	"create"	"country"	"economy"	"economy"	"growth"	"create"	"economic"
[10,]	"economy"	"help"	"businesses"	"global"	"plan"	"growth"	"measures"
[11,]	"growth"	"canadian"	"provide"	"plan"	"term"	"economy"	"tax"
[12,]	"ensure"	"funding"	"tax"	"tax"	"business"	"ensure"	"commitment"
[13,]	"world"	"advantage"	"jobs"	"country"	"stability"	"world"	"skills"
[14,]	"businesses"	"fiscal"	"families"	"spending"	"taxes"	"businesses"	"support"
[15,]	"investments"	"plan"	"global"	"help"	"businesses"	"investments"	"world"
[16,]	"program"	"world"	"support"	"action"	"countries"	"canadian"	"building"
[17,]	"canadian"	"health"	"projects"	"economic"	"financial"	"creating"	"country"
[18,]	"creating"	"children"	"recession"	"recession"	"fiscal"	"program"	"infrastructure"
[19,]	"security"	"families"	"relief"	"business"	"provide"	"security"	"prosperity"
[20,]	"age"	"helping"	"billion"	"create"	"support"	"age"	"term"

	Lib_1966.txt	Lib_1968.txt	Lib_1969.txt	Lib_1970.txt	Lib_1971.txt	Lib_1972.txt	Lib_1973.txt
[1,]	"tax"	"tax"	"million"	"million"	"tax"	"tax"	"tax"
[2,]	"increase"	"income"	"fiscal"	"fiscal"	"income"	"income"	"income"
[3,]	"economic"	"canada"	"tariff"	"credit"	"system"	"economy"	"budget"
[4,]	"capital"	"fiscal"	"tax"	"budgetary"	"capital"	"canada"	"government"
[5,]	"fiscal"	"million"	"economy"	"growth"	"canadian"	"canadians"	"provinces"
[6,]	"canada"	"canadian"	"reductions"	"budget"	"government"	"government"	"cent"
[7,]	"cent"	"percent"	"increase"	"economic"	"cent"	"measures"	"measures"
[8,]	"income"	"provinces"	"prices"	"increase"	"reform"	"growth"	"inflation"
[9,]	"rate"	"rates"	"budget"	"prices"	"rate"	"canadian"	"increase"
[10,]	"government"	"board"	"budgetary"	"revenues"	"corporations"	"cost"	"jobs"
[11,]	"million"	"increase"	"costs"	"government"	"changes"	"jobs"	"prices"
[12,]	"economy"	"proposed"	"price"	"increases"	"economy"	"increase"	"system"
[13,]	"investment"	"budget"	"cent"	"net"	"gains"	"provide"	"federal"
[14,]	"time"	"economic"	"government"	"rate"	"taxpayers"	"fiscal"	"basic"
[15,]	"expenditures"	"companies"	"canada"	"cent"	"canada"	"manufacturing"	"economic"
[16,]	"revenues"	"current"	"economic"	"hear"	"personal"	"tonight"	"growth"
[17,]	"provincial"	"revenue"	"effect"	"1969"	"taxes"	"time"	"rate"
[18,]	"national"	"total"	"canadian"	"197071"	"million"	"cent"	"reduction"
[19,]	"account"	"insurance"	"country"	"canada"	"business"	"million"	"taxes"
[20,]	"budget"	"reserves"	"october"	"cash"	"canadians"	"propose"	"time"

	Lib_1974.txt	Lib_1975.txt	Lib_1976.txt	Lib_1977.txt	Lib_1978.txt	Lib_1981.txt	Lib_1983.txt
[1,]	"tax"	"government"	"government"	"tax"	"tax"	"tax"	"recovery"
[2,]	"cent"	"tax"	"cent"	"business"	"cent"	"rates"	"tax"

[3,]	"income"	"canada"	"tax"	"inflation"	"increase"	"inflation"	"canadians"
[4,]	"canada"	"increase"	"rate"	"income"	"income"	"canadians"	"government"
[5,]	"measures"	"oil"	"fiscal"	"policy"	"rate"	"income"	"special"
[6,]	"government"	"federal"	"growth"	"credit"	"credit"	"government"	"investment"
[7,]	"increase"	"economy"	"unemployment"	"government"	"investment"	"economic"	"million"
[8,]	"inflation"	"cost"	"increase"	"prices"	"measures"	"growth"	"private"
[9,]	"rate"	"prices"	"canada"	"investment"	"taxes"	"rate"	"federal"
[10,]	"federal"	"cent"	"inflation"	"capital"	"canada"	"budget"	"program"
[11,]	"prices"	"price"	"economic"	"cent"	"canadians"	"cent"	"canada"
[12,]	"canadians"	"costs"	"policy"	"economic"	"costs"	"restraint"	"budget"
[13,]	"economy"	"income"	"canadian"	"growth"	"government"	"development"	"employment"
[14,]	"measure"	"measures"	"changes"	"measures"	"growth"	"billion"	"projects"
[15,]	"canadian"	"rate"	"economy"	"rate"	"sales"	"lower"	"canadian"
[16,]	"million"	"inflation"	"program"	"budget"	"trade"	"federal"	"billion"
[17,]	"savings"	"employment"	"firms"	"canadians"	"budget"	"reduce"	"cent"
[18,]	"fiscal"	"public"	"prices"	"unemployment"	"competitive"	"system"	"sector"
[19,]	"international"	"economic"	"income"	"canada"	"development"	"canada"	"increase"
[20,]	"propose"	"million"	"price"	"canadian"	"million"	"changes"	"development"

	Lib_1984.txt	Lib_1990.txt	Lib_1991.txt	Lib_1992.txt	Lib_1994.txt	Lib_1995.txt	Lib_1996.txt
[1,]	"tax"	"government"	"government"	"budget"	"government"	"government"	"government"
[2,]	"government"	"billion"	"budget"	"government"	"budget"	"budget"	"canadians"
[3,]	"canadians"	"deficit"	"economic"	"canadians"	"business"	"canadians"	"tax"
[4,]	"business"	"economic"	"canadians"	"canada"	"canadians"	"tax"	"system"
[5,]	"economic"	"spending"	"canada"	"cent"	"tax"	"system"	"budget"
[6,]	"budget"	"tax"	"tax"	"tax"	"jobs"	"billion"	"support"
[7,]	"cent"	"<U+0095>"	"cent"	"growth"	"reform"	"canada"	"canada"
[8,]	"jobs"	"canada"	"speech"	"economic"	"canada"	"spending"	"economy"
[9,]	"provide"	"debt"	"spending"	"speech"	"speech"	"fiscal"	"time"
[10,]	"private"	"growth"	"plan"	"economy"	"system"	"country"	"country"
[11,]	"growth"	"economy"	"inflation"	"spending"	"social"	"provinces"	"health"
[12,]	"pension"	"program"	"recovery"	"cut"	"program"	"ensure"	"fiscal"
[13,]	"investment"	"canadians"	"growth"	"business"	"spending"	"canadian"	"future"
[14,]	"sector"	"inflation"	"federal"	"capital"	"fiscal"	"cent"	"jobs"
[15,]	"security"	"plan"	"fiscal"	"inflation"	"growth"	"reform"	"parents"
[16,]	"support"	"programs"	"public"	"action"	"minister"	"deficit"	"spending"
[17,]	"income"	"1984"	"programs"	"income"	"canadian"	"program"	"billion"
[18,]	"recovery"	"budget"	"debt"	"million"	"deficit"	"unemployment"	"child"
[19,]	"assistance"	"cent"	"lower"	"1992"	"economic"	"business"	"income"
[20,]	"canadian"	"control"	"program"	"billion"	"insurance"	"insurance"	"pension"

	Lib_1997.txt	Lib_1998.txt	Lib_1999.txt	Lib_2000.txt	Lib_2001.txt	Lib_2003.txt	Lib_2004.txt
[1,]	"canada"	"canadians"	"health"	"tax"	"canada"	"canada"	"canada"
[2,]	"speaker"	"canada"	"canadians"	"speaker"	"budget"	"government"	"government"
[3,]	"government"	"speaker"	"budget"	"budget"	"speaker"	"budget"	"budget"
[4,]	"canadians"	"tax"	"canada"	"canadians"	"provide"	"million"	"canadians"
[5,]	"budget"	"education"	"speaker"	"canada"	"billion"	"health"	"speaker"
[6,]	"children"	"budget"	"tax"	"economy"	"canadians"	"canadians"	"health"
[7,]	"health"	"government"	"care"	"cent"	"economy"	"economic"	"communities"
[8,]	"million"	"students"	"billion"	"government"	"million"	"tax"	"education"
[9,]	"time"	"country"	"research"	"children"	"research"	"canadian"	"billion"
[10,]	"country"	"time"	"government"	"plan"	"security"	"care"	"provide"
[11,]	"tax"	"opportunity"	"million"	"time"	"canadian"	"economy"	"canadian"
[12,]	"future"	"access"	"debt"	"canadian"	"government"	"fiscal"	"social"
[13,]	"jobs"	"debt"	"system"	"economic"	"cent"	"billion"	"economy"
[14,]	"deficit"	"children"	"country"	"health"	"economic"	"growth"	"people"
[15,]	"education"	"support"	"knowledge"	"income"	"people"	"support"	"children"
[16,]	"provide"	"canadian"	"canadian"	"country"	"time"	"improve"	"families"
[17,]	"billion"	"child"	"resources"	"taxes"	"health"	"five"	"future"
[18,]	"support"	"provide"	"announced"	"million"	"infrastructure"	"2003"	"million"
[19,]	"system"	"financial"	"time"	"child"	"funding"	"accountability"	"care"
[20,]	"economy"	"skills"	"benefit"	"debt"	"support"	"investments"	"economic"

	Lib_2005.txt
[1,]	"canada"
[2,]	"budget"
[3,]	"canadians"
[4,]	"billion"
[5,]	"care"

```
 [6,] "tax"
 [7,] "health"
 [8,] "federal"
 [9,] "canadian"
[10,] "five"
[11,] "government"
[12,] "million"
[13,] "world"
[14,] "commitment"
[15,] "economic"
[16,] "funding"
[17,] "help"
[18,] "people"
[19,] "provide"
[20,] "aboriginal"
```

TABLE-3 : The top 20 words in each of the budget speeches

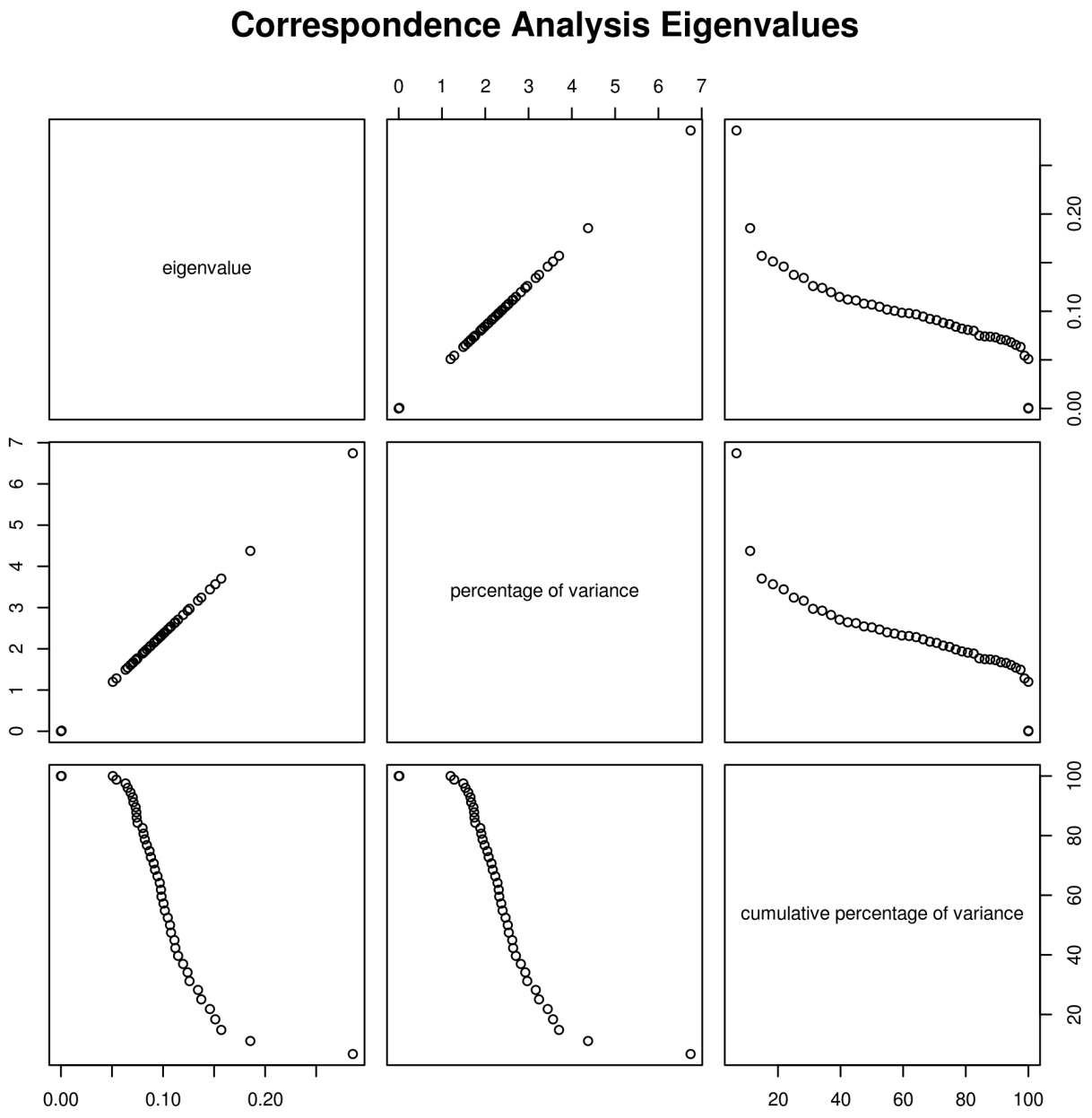

FIGURE 1 : Plot of the correspondence analysis eigenvalues, showing the relationship between the values, percentage of variance, and the cumulative percentage of varience.

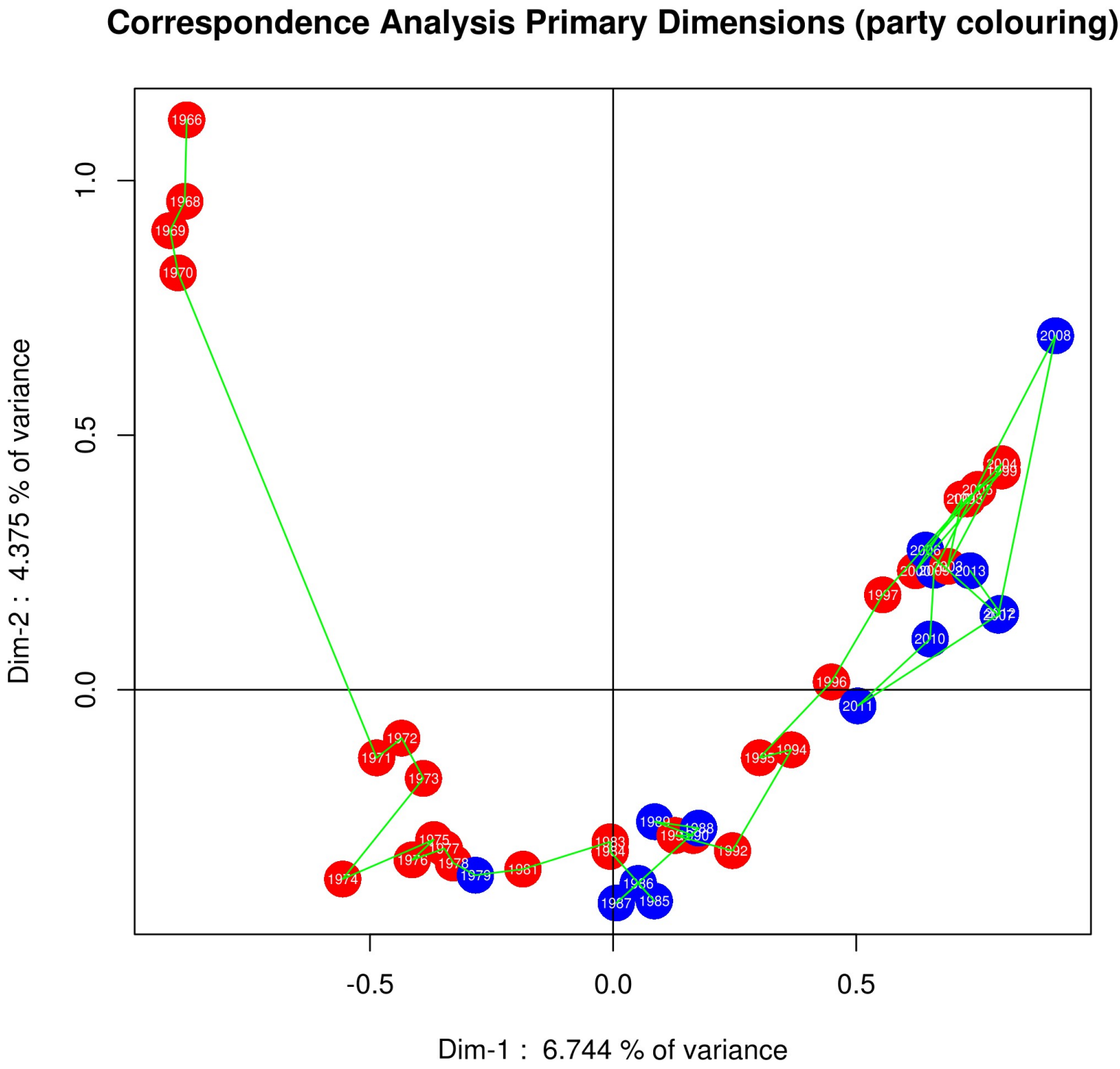

FIGURE 2 : Plot of the speeches using the first 2 dimensions of the analysis. The colours indicate the party (RED = Liberal, BLUE = PC or Conservative). The green line is used to link one year to the next, and is used to illustrate the "drift" of the tone of the speeches from one year to the next.

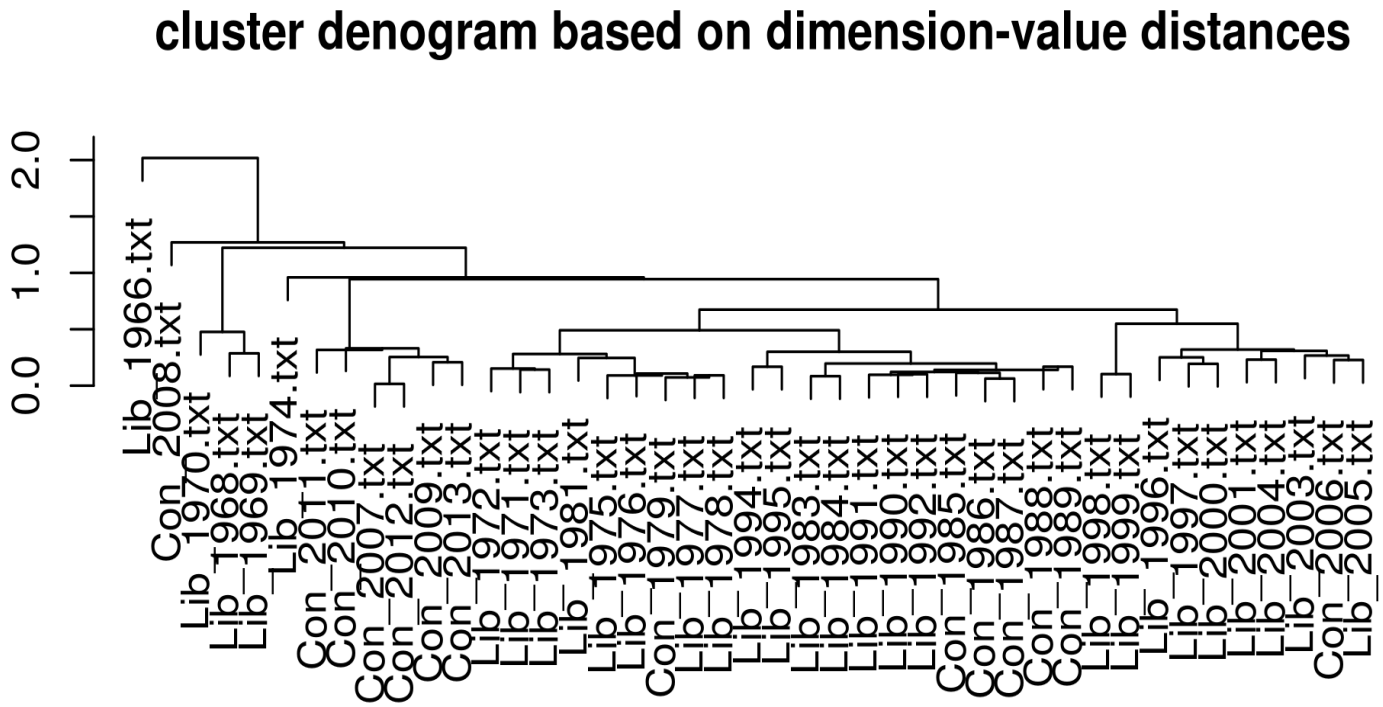

FIGURE 3 : A cluster analysis of the speeches, based on a distance calculated from the dimension data.

PROGRAM-1 “AnalyzeBudgetSpeeches.R”

```
#
# Analyse Budget Speeches
#
# April 09/13 by bcg
#   - added saving to files for text and plot output
#
# April 05/13 by bcg
#   - first cut
#

# set working and data directories
setwd("/analyzeBudgetSpeeches")

dataSet <- "Federal"
#dataSet <- "Federal_Con"
#dataSet <- "Federal_Lib"
#dataSet <- "Ontario"

dataDir <- paste(getwd(),"/", dataSet,sep="")
#dataDir <- paste(getwd(),"/Federal", sep="")
#dataDir <- paste(getwd(),"/Federal_Con", sep="")
#dataDir <- paste(getwd(),"/Federal_Lib", sep="")

# file name for text output
analysisFileName <- paste("Diagrams/", dataSet, "_analysis.txt",sep="")


# load the analysis utilities
source("ExploratoryCode/AnalysisUtilities.R")


# create a term document matrix from the corpus
budget.TDM <- createFilteredCorpus(dataDir)


# do correspondence analysis on the TDM
retVal <- correspondenceAnalysis(budget.TDM)
budget.CA <- retVal[[1]]
budget.CA.desc <- retVal[[2]]
budget.CA.names <- retVal[[3]]


sink(analysisFileName, append=FALSE)
textSummary <- paste("Total number of documents analysed in dataset ",
```

```
dataSet, " : ", length(budget.CA.names[[1]]))
print(textSummary)
sink()


# examine eigenvalues
sink(analysisFileName, append=TRUE)
#sink("Diagrams/analysis.txt", append=FALSE)
print(budget.CA$eig, digits=4)
sink()

plotFileName <- paste("Diagrams/", dataSet,
"_CA_eigenvalues.bmp",sep="")
bmp(plotFileName, width=7, height=7, units="in", res=300)
plot(budget.CA$eig, main="Correspondence Analysis Eigenvalues")
dev.off()


# do cluster denogram and plot it
# the "centroid" clustering seems to produce nicer clusters
# budget.deno <- distanceDenogram(budget.CA)
# budget.deno <- distanceDenogram(budget.CA, distMethod="manhattan")
# budget.deno <- distanceDenogram(budget.CA, distMethod="manhattan",
hclustMethod="single")
budget.deno <- distanceDenogram(budget.CA, distMethod="euclidean",
hclustMethod="centroid")

plotFileName <- paste("Diagrams/", dataSet,
"_distanceDenogram.bmp",sep="")
bmp(plotFileName, width=7, height=4, units="in", res=300)
plotDistanceDenogram(budget.deno)
dev.off()

#plotDistanceDenogram(budget.deno, 8)


# what are the most popular words?
popWords <- mostPopularWords(budget.TDM, sort(budget.CA.names[[1]]), 20)
sink(analysisFileName, append=TRUE)
popWords
sink()

# plot the primary dimensions of the Correspondence Analysis
plotFileName <- paste("Diagrams/", dataSet,
"_CA_partycolours.bmp",sep="")
bmp(plotFileName, width=7, height=7, units="in", res=300)
```

```
plotDimDescData(budget.CA, budget.CA.desc, budget.CA.names,
progressLines=TRUE)
dev.off()
```

PROGRAM-2 “AnalysisUtilities.R”

```
#
# Analysis Utilities
#   - a series of functions relating to the analysis of the text files
#
# April 20/13 by bcg
#  - fix to createFilteredCorpus() to read as UTF-8
# April 04/13 by bcg
#   - first cut


# =====  load required libraries  ================
library(tm)
library(wordcloud)
library(FactoMineR)


# ================================================
# =====  createFilteredTDM()  ====================
# ================================================
#
# load all the data files into a single corpus,
# then apply a standard set of filters, then
# calculate a term document matrix.
# INPUT   : sourceDirectory (fully-qualified)
# RETURN  : term document matrix
#
createFilteredCorpus <- function(sourceDirectory)
{
  # load all the files into a single corpus
  b <- Corpus(DirSource(sourceDirectory, encoding="UTF-8"))

  # filter out stuff that just confuses the analysis
  # white space, stop words punctuation
  b <- tm_map(b, tolower)               #Changes case to lower case
  b <- tm_map(b, stripWhitespace)
  b <- tm_map(b, removeWords, stopwords("english"))
  b <- tm_map(b, removePunctuation)

  # create a term document matrix
  tdm <- TermDocumentMatrix(b)

  return(tdm)

}
```

```
# ================================================
# =====  mostPopularWords()  =====================
# ================================================
#
# From a given term document matrix, calculate
# the most popular words for each document.
# Actually picks the 70th percentile, and then
# returns the top_N of that.  Typically used to
# the tope 10-20 words from a large document, so
# that algorithm should be fine.
#
# INPUT   : term document matrix
#           vector of file names
#           number of top words
# RETURN  : matrix of most popular words, arranged
#           with each row as a different document
#
mostPopularWords <- function(tdm, fileList, numWords)
{
  if (numWords < 1)
  {
    print("ERROR : numWords too small")
    return
  }

  # create matrix for the most popular words
  popularWords <- matrix(nrow=numWords, ncol=length(fileList))
  colnames(popularWords)<-fileList
  colCount <- 1

  for (fileName in fileList)
  {
    termFrequency <-
rowSums(as.matrix(tdm[,tdm$dimnames$Docs==fileName]))
    termFrequency.matrix <-
as.matrix(rowSums(as.matrix(tdm[,tdm$dimnames$Docs==fileName])))
    wf = rowSums(as.matrix(termFrequency))
    wf = rowSums(termFrequency.matrix)
    termFrequency2 = termFrequency.matrix[wf>quantile(wf,probs=.7),]
    v1<- sort(termFrequency2,decreasing=TRUE)

    popularWords[, colCount] <- names(v1[1:numWords])
    colCount <- colCount + 1
```

```
  }

  return (popularWords)

}



# ================================================
# =====  barplotMostPopularWords()  ==============
# ================================================
#
# From a given term document matrix, calculate
# the most popular words for each document, then
# create a barplot for each document.
# Actually picks the 70th percentile, and then
# returns the top_N of that.  Typically used to
# the tope 10-20 words from a large document, so
# that algorithm should be fine.
#
# INPUT   : term document matrix
#           number of top words
# RETURN  : -
#
barplotMostPopularWords <- function(tdm, numWords)
{
  if (numWords < 1)
  {
    print("ERROR : numWords too small")
    return
  }


  for (fileName in fileList)
  {
    termFrequency <-
rowSums(as.matrix(tdm[,tdm$dimnames$Docs==fileName]))
    termFrequency.matrix <-
as.matrix(rowSums(as.matrix(tdm[,tdm$dimnames$Docs==fileName])))
    wf = rowSums(as.matrix(termFrequency))
    wf = rowSums(termFrequency.matrix)
    termFrequency2 = termFrequency.matrix[wf>quantile(wf,probs=.7),]
    v1<- sort(termFrequency2[1:numWords],decreasing=TRUE)

    barplot(v1, las=2, main=fileName, horiz=TRUE)
  }
```

```
}


# ================================================
# =====  correspondenceAnalysis()  ===============
# ================================================
#
# From a given term document matrix, calculate
# the correspondence analysis.  Then calculate
# it's description, and from that extract the
# names of the documents.
#
# INPUT   : term document matrix
# RETURN  : row and column points factor map.
#           description analysis
#           names of the documents in the description analysis
#
correspondenceAnalysis <- function(tdm)
{
  # calculate the correspondence analysis
  corrAn=CA(as.matrix(tdm),graph=FALSE)

  # corr. anal. description analysis
  dd <- dimdesc(corrAn)

  # names
  docNames<-dimnames(dd$`Dim 1`$col)

  retValue <- list(corrAn, dd, docNames)

}



# ================================================
# =====  distanceDenogram()  =====================
# ================================================
#
# From a given correspondence analysis calculate
#  distances and use those to do a hierarchial
# cluster analysis.
#
# INPUT   : correspondence analysis
#           dist() method, default = euclidian
```

```
#           hclust() method, default = ward
# RETURN  : object of class hclust
#
distanceDenogram <- function(corrAnalysis, distMethod="euclidian",
hclustMethod="ward")
{
  d <- dist(corrAnalysis$col$coord, method=distMethod)
  denFit <- hclust(d, method=hclustMethod)

  return(denFit)

}



# ================================================
# =====  plotDistanceDenogram()  =================
# ================================================
#
# Plot a calculated hierarchial cluster analysis.
#
# INPUT   : object of class hclust
#           number of clusters
#           (used to cut tree into groups, if >1)
# RETURN  : -
#
plotDistanceDenogram <- function(denFit, numClusters=0)
{
  plot(denFit, xlab="", ylab="",
          main="cluster denogram based on dimension-value distances",
          sub="")

  # cut the tree into clusters
  # then draw red borders around the clusters
  if (numClusters > 1)
  {
    groups <- cutree(denFit, k=numClusters)
    rect.hclust(denFit, k=numClusters, border="red")
  }

}     # =====  end of plotDistanceDenogram()  =====



# ================================================
# =====  plotDimDescData()  ======================
```

```
# ================================================
#
# Plot the Dim Description data .
# Colour the data according to party
# (Lib=red, Cons=blue)
#
# INPUT   : corr. analysis object
#           dim. desc. object
#           file_names dataframe
#           draw year-year trajectory line (T/F)
# RETURN  : -
#
plotDimDescData <- function(rei_ca, dd, rnm, progressLines=FALSE)
{
  # plot as points
  # colour by party
  d1<-format(rei_ca$eig[[2]][1], digits=4)
  d2<-format(rei_ca$eig[[2]][2], digits=4)
  partyColour <- substr(rnm[[1]],1,1)
  partyColour<-sub("C","blue",partyColour)
  partyColour<-sub("L","red",partyColour)
  xLabel <- paste("Dim-1 : ", d1, "% of variance")
  yLabel <- paste("Dim-2 : ", d2, "% of variance")
  plot(dd$`Dim 1`$col[rnm[[1]],], dd$`Dim 2`$col[rnm[[1]],],
       #     col=1:8,
       col=partyColour,
       cex=3,
       #     cex=5.6,
       pch=16,
       type="p",
       xlab=xLabel, ylab=yLabel,
       main="Correspondence Analysis Primary Dimensions (party
colouring)")
  # add axis lines
  abline(0,0)
  abline(0,10000)
  # put the years into the dots
  # NOTE : this currently assumes a fixed format for the file names!
(xxx_YYYY.txt)
  labels <- as.vector(substr(rnm[[1]], 5, 8))
  text(dd$`Dim 1`$col[rnm[[1]],],
       dd$`Dim 2`$col[rnm[[1]],],
       labels,
       col='white',
       cex=.5)
```

```
  # add lines showing progress-path over time
  # first we need to sort the list of names by year
  # (not so easy, since format is XXX_YYYY.txt)
  if (progressLines == TRUE)
  {
    bNames <- matrix(ncol=2, nrow=length(rnm[[1]]))
    bNames[,1]<-rnm[[1]]
    bNames[,2]<-substr(rnm[[1]], 5, 8)
    sortedNames <- bNames[order(bNames[,2])]

#     lines(dd$`Dim 1`$col[sort(rnm[[1]]),],
#           dd$`Dim 2`$col[sort(rnm[[1]]),],
    lines(dd$`Dim 1`$col[sortedNames,],
          dd$`Dim 2`$col[sortedNames,],
          col='green',
          cex=3)
  }

}     # =====  end of plotDimDescData()  =====
```

PROGRAM-3 : “CleanUpText.R”

```
#
# FILTER TEXT FILES
#
# Gets rid of funky characters (x96, x97, x98, x99) that sometimes
# get put into files scraped from web page or extracted from a PDF.
#
# These funky characters cause problems for the tm_map() operations.
#
#
# This program takes all the text files found in the source directory
# and replaces those characters with a '-' character.  It then writes #
the filtered data out to the target directory.
#
# April 20/13 by bcg
#   - added filtering of xe9 and xe8 (replace with "e")
#   - replaced "\" with "\\" so things would work under Linux
# April 12/13 by bcg
#   - added filtering of x93,x94 (replaced with " symbol)
#
# April 11/13 by bcg
#   - first cut
#

#library(stringr)

# set working and data directories
setwd("/analyzeBudgetSpeeches")

# force the local to be common amoung any and all machines
Sys.setlocale('LC_ALL','C')

# define the dataset to be used, and where to put filtered data
//dataSet <- "Ontario_Raw"
//dataTarget <- "Ontario"
dataSet <- "Federal0"
dataTarget <- "Federal"

dataDir <- paste(getwd(),"/", dataSet,sep="")
targetDir <- paste(getwd(),"/", dataTarget,sep="")

# get list of all the files in the source directory
fileList <- list.files(dataDir)

# process each file in the source directory
```

```
for (fileName in fileList)
{
  fullName <- paste(dataDir,"/", fileName, sep="")

  print(paste("Working on ", fullName))

  # load the text from the file
  fileIn <- file(fullName)
  inData <- readLines(fileIn)
  close(fileIn)

  # using scan() creates a vector of words (ie. destroys original
structure)
  #  inData <- scan(fullName, what="character")

  # show if there any of the extra characters
  print(length(grep('\\xe8', inData)))
  print(length(grep('\\xe9', inData)))
  print(length(grep('\\x93', inData)))
  print(length(grep('\\x94', inData)))
  print(length(grep('\\x96', inData)))
  print(length(grep('\\x97', inData)))
  print(length(grep('\\x98', inData)))
  print(length(grep('\\x99', inData)))

  # if there are any funky characters to deal with
  if (   (length(grep('\\x96', inData)) > 0) ||
         (length(grep('\\xe8', inData)) > 0) ||
         (length(grep('\\xe9', inData)) > 0) ||
         (length(grep('\\x93', inData)) > 0) ||
         (length(grep('\\x94', inData)) > 0) ||
         (length(grep('\\x97', inData)) > 0) ||
         (length(grep('\\x98', inData)) > 0) ||
         (length(grep('\\x99', inData)) > 0) )
  {
    # replace 'em
    filteredData <- sub('\\x96', '-', inData)
    filteredData <- sub('\\x97', '-', filteredData)
    filteredData <- sub('\\x98', '-', filteredData)
    filteredData <- sub('\\x99', '-', filteredData)
    filteredData <- sub('\\x99', '-', filteredData)
    filteredData <- sub('\\x93', '"', filteredData)
    filteredData <- sub('\\x94', '"', filteredData)
    filteredData <- sub('\\xe8', 'e', filteredData)
    filteredData <- sub('\\xe9', 'e', filteredData)
```

```
    # see if there any of the extra characters left after filtering
    # grep('\x96', filteredData)
    # grep('\x97', filteredData)
    # grep('\x98', filteredData)
    # grep('\x99', filteredData)

    # write the filtered data to the target directory
    targetName <- paste(targetDir,"/", fileName, sep="")
    fileOut <- file(targetName, encoding = "UTF-8")
    writeLines(filteredData, fileOut)
    close(fileOut)

    outStr <- paste("   ", fileOut)
    print(outStr)
  }
  else
  {
    print("   nothing to filter")

    # write the original data to the target directory
    targetName <- paste(targetDir,"/", fileName, sep="")
    fileOut <- file(targetName)
    writeLines(inData, fileOut)
    close(fileOut)

    outStr <- paste("   ", fileOut)
    print(outStr)
  }

}     # =====  end of for()
```