Text Mining Internet Discussion Groups an Automotive Case Study

John Wallace, Business Researchers, Inc., San Francisco, CA

ABSTRACT

Public domain information can provide a unique source for data collection for both academic and corporate research. This paper explains how systematically analyzing the USENET can benefit companies, what the modeling process entailed and the steps required to clean the data. The USENET data serves as the platform to discuss SAS Text Miner and the process involved in using it and other analytical tools for text mining. The pre-processing of data, exploratory analysis and development of synonym and stop lists is covered. The final model is a hierarchy of Expectation Maximization Clustering models that total over 100 clusters.

INTRODUCTION

This paper uses public data from the USENET to demonstrate the ability of commercial text mining software free form text. Our analysis had a modeling objective of categorizing postings into useful groups. The definition of useful was that the group would contain postings that were relatively homogenous when profiled by an analyst. Although the homogeneity measure was somewhat subjective, the percent of documents containing the most frequent word was one quick indication.

This data is used for demonstration purposes. Several modeling projects on corporate databases have included analysis of customer service call notes, warranty claims, technical call center notes and insurance claims.

DATA SOURCE

There is no requirement on the text's format in each database; any combination of characters is allowed. The text is from the USENET group alt.autos.subaru. The dataset contains postings from September 2004 – April 2005. That period yielded about 8,000 posts.

The language used to discuss cars on the Internet differs significantly from English prose. The postings are loaded with slang, abbreviations, technical jargon, misspellings, etc. The quality of the text could be rated as very dirty to say the least.

WORKFLOW

The Early Warning Process begins by collecting the data for a modelset (see Figure 1). Modelsets included all of the records (a minimum of three months) in the database about a specific car. We chose to separate the statistical models by the three types of data listed above, as well as car model and the model year.

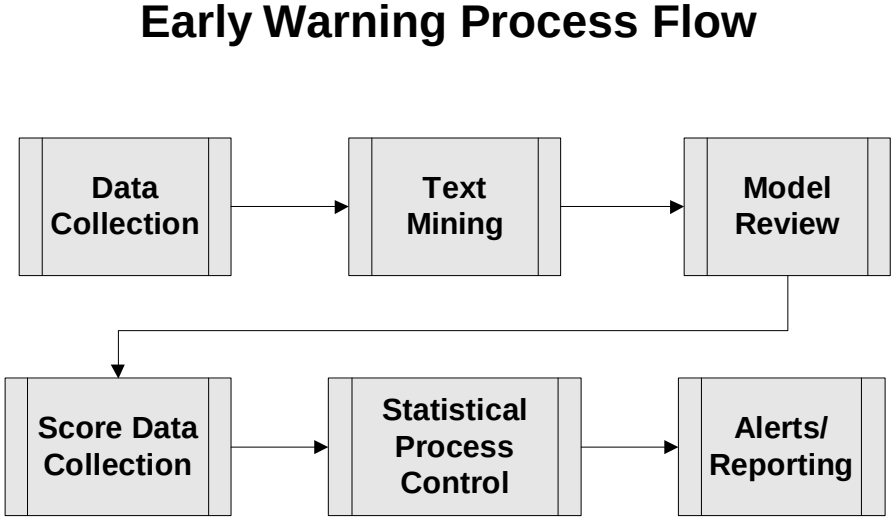

Figure 1.

Once the clustering model has completed the process described below, the model is put into production and it begins to score new data. On a weekly basis, new data is scored and assigned to the best cluster. The week's assignments are aggregated and analyzed using SAS/QC. Any departure from the expected aggregate values is flagged and an alert is included in the week's alert email.

CLUSTERING

PROCESS

Figure 2 shows the flow of documents through the text mining process. The step "Term by Document Matrix" is where words are converted to numbers (see *Getting Started with Text Miner* for more detail). The size and sparseness of the term by document matrix is driven by the size and complexity of the document collection.

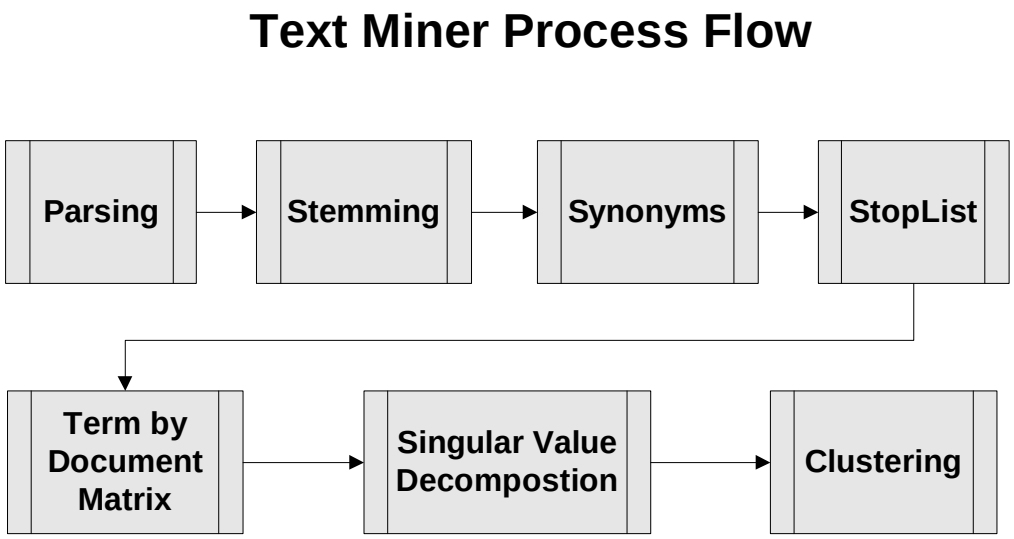

Figure 2.

The inputs to the clustering algorithm are the outputs from the Singular Value Decomposition: the SVD document vectors. The number of vectors needed to best approximate that matrix typically ranges from 10 to 100. In Text Miner the parameter *resolution* is applied to decide how many of the vectors to use for clustering.

In this project, up to 70 dimensions were passed to the clustering algorithm as input. A space with so many dimensions is very empty; the observations are pushed to the edges of the projected space. In essence the clustering algorithm is identifying intersections on the edges where there are groups of observations. Figure 3 depicts what this space may look like. Clusters of varying sizing and varying distances from one another inhabit the edges of the input space.

Clusters on the Outer Edges of the Input Space

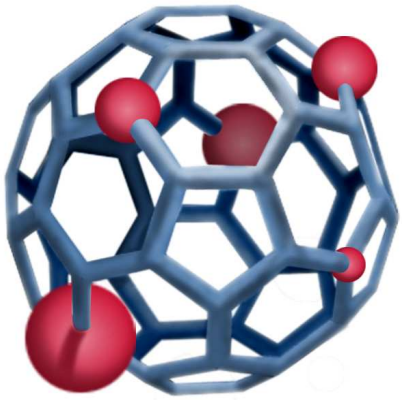

Figure 3.

Some experimentation during early development with both K-means (PROC FASTCLUS) and Expectation-Maximization (PROC EMCLUS) clustering led to the selection of Expectation-Maximization clustering for all models. The decision was based on the quality of the models as defined in the Application section below.

Once the statistical modeler has built an initial model, the modeler presents the clusters to the domain expert: the auto analyst. Lastly, after the auto analyst makes any changes, the final clustering model becomes the baseline knowledge about all of the free-form text for that car.

APPLICATION

The most general objective of clustering is to form groups of similar records. The objective of grouping documents together in a meaningful fashion can only be met when the auto analyst finds the clusters useful. In this case, the usefulness of a cluster is subjective but focuses on the homogeneity of the documents within it. The domain expert assesses the homogeneity of a cluster as to how well that cluster describes a particular engineering problem. For example, if every document in a cluster refers to the headliner, it is more homogenous then a cluster with documents about the bumper and the headliner.

Since a document collection may have hundreds of different topics discussed in a one-year period, tens or hundreds of clusters may be necessary in order to achieve homogeneity among documents in a cluster. Based on the modeler's domain expertise, clustering models were tuned for homogeneity using two techniques: subclustering and merging clusters.

Subclustering can be defined as using the documents assigned to a cluster in a first model as the modelset for a subsequent model. The subcluster model had a much smaller term by document matrix, and therefore allowed documents that were once viewed as similar to be grouped separately. One particular cluster contained some related topics and one could understand how they were drawn together. The example was remote entry, alarm and remote start. These are all systems that are controlled by the car's remote control. Our subclustering step separated these out into the 3 distinct groups. Lastly, if the model identified a cluster about cars with a dead battery and another cluster about cars being jump started, it was easy to merge the two clusters together outside of Text Miner.

TEXT MINING UTILITIES

At first glance, text mining seems to contradict the adage that data mining is 80% data preparation and 20% modeling. The raw text itself is the input to the algorithms. However, two other important inputs end up taking considerable time to develop: the synonym list and stop list. Just as in data mining the fitting of lines to the data is arguably less important than understanding the problem and creating useful input variables, in text mining creating these lists is more important than the selection of clustering algorithms. These lists are the heart (or brains) of any Text Miner model. Due to the number of abbreviations and auto-specific terms, this project's synonym list has grown to over 30,000 entries.

During the course of developing the first models, we opted to build some utilities to extend the functionality of Text Miner using Base SAS and SAS Macro. Several of these extensions were related to synonym and stop list creation.

A partial list of the utilities developed follows:

1. Data Preprocessor. Particularly dirty data may require the removal of unwanted characters using rules other than those employed by Text Miner. The portion of this macro shown below addresses the conditional removal of a slash and removes incomplete records. The slash is used to create the word "A/C" but may also be used to join two separate words like "repaired/installed". The conditional removal is necessary, as A/C cannot become A C. If "repaired/installed" is left alone, it means neither "repaired" nor "installed", but a new third term to Text Miner. The algorithm in this macro works in this order:

    a. Count up to the first three words in the string.
    b. If there are less than three, delete the record.
    c. Look for a slash at the beginning and end of the string and remove it.
    d. From the beginning of the string, identify the first occurrence of a slash.
    e. Count the number of characters in front of the slash.
    f. If three or more characters, reconstruct the string with a space in place of the slash.
    g. If the string proceeding the slash is shorter (i.e. A/C), skip and continue searching in the rest of the string.

```
%macro RemoveUnwantedCharacters(DsIn=, TextField=);
…
Code removed here
…
data &DsIn (drop=location location2 count i);
set &DsIn;
wordcount=0;
do i=1 to 3;
  if scan(&TextField, i, " ") ne "" then wordcount+1;
end;
if wordcount <=2 then delete;
if substr(reverse(&TextField),1,1)="/" then
  &TextField=reverse(substr(reverse(&TextField),2));
if substr(&TextField,1,1)="/" then
  &TextField=substr(&TextField,2);
count=1;
location=index(&TextField,"/");
if location > 0 then do until (location2=0);
if length(scan(reverse(scan(&TextField, count, "/")),1," ")) > 2 then do;
  &TextField=trim(substr(&TextField, 1, location-1))
  !!" "!!trim(substr(&TextField, location+1));
end;
else do;
  count+1;
end;
  location2=index(substr(&TextField,location+1),"/");
  location=location+location2;
end;
run;

%mend;
```

2. Dataset Extractor. This routine extracts term lists from Text Miner into Excel for human review. On the way into Excel, these lists were then processed by other SAS macros not detailed here.
3. History Recorder. This step maintains a master term list so that only new words (i.e. words that have not been seen before) in new documents are reviewed.
4. Text Helper. The human decision process of adding terms to the synonym and/or stop list is facilitated with Excel-based (Visual Basic) macros.
5. Synonym Integrity Checker. A macro analyzes the integrity of the synonym list in order to remove inconsistencies (i.e. a term that has multiple parents) that can have an unpredictable impact on Text Miner. Any exceptions are presented to the operator for correction and corrections are integrated into the synonym and stop lists. These checks are required often because the synonym and stop lists are updated each time a new model is built.
6. Nuisance Record Suppressor. Adding terms to the stop list removes the term from cluster definition. However sometimes it is meaningful to remove the whole document. Whatever types of records were suppressed in the modeling stage also need to be suppressed during the scoring process.
7. Model Reporter. Having the top terms for every cluster and subcluster in a file became very useful when reviewing models with the domain experts.

SCORING

The main facility for scoring provided in Text Miner on SAS 8.2 is scoring from the Enterprise Miner interface. However, all of the necessary score code to score new data in batch is available in the score node. Some of the modeling decisions such as subclustering and suppressing observations called for careful creation of Enterprise Miner diagrams and several enhancements to the Text Miner score code. After some experimentation there was success in gathering the score code from a diagram with multiple models and scoring data outside of the EM interface. Simplifying the process of deploying the score code was critical since nearly 100 clustering models would be put into production.

MONITORING

Building clustering models was a means to an end: to monitor incoming text data. The monitoring process was tasked with finding potential problems and to alert auto analysts about them. The initial set of monitoring processes and reports are described below.

CHANGE-IN-SIZE ALERTS

The “change-in-size alert” monitors cluster sizes on a weekly basis and signals abnormal growth by utilizing the p-charting capability of SAS QC’s PROC SHEWHART. In our case the proportion of total records in a specific cluster took the place of the traditional proportion nonconforming and the week’s total number of records became the varying sample total.

PROC SHEWHART calculates the appropriate control limits (3sigma) from the data depending on the variance of the clusters’ proportion and total sample size. The procedure also saves calculated limits and reads them back in during subsequent weeks. Analysts are alerted about any cluster for which one or more weeks saw the cluster’s proportion of total records above the upper control limit. In addition, if five or more periods in a row are trending in the same direction, an alert is also generated.

Although this is not the classic application of a p-chart, its capabilities have proven solid. One concern early on was that the proportion that was being charted was a percent of total, not a simple proportion conforming. The zero-sum nature of a percent of total may have been problematic but the large number of clusters (>100) softened the impact of one cluster’s growth reducing a constant cluster’s proportion of total.

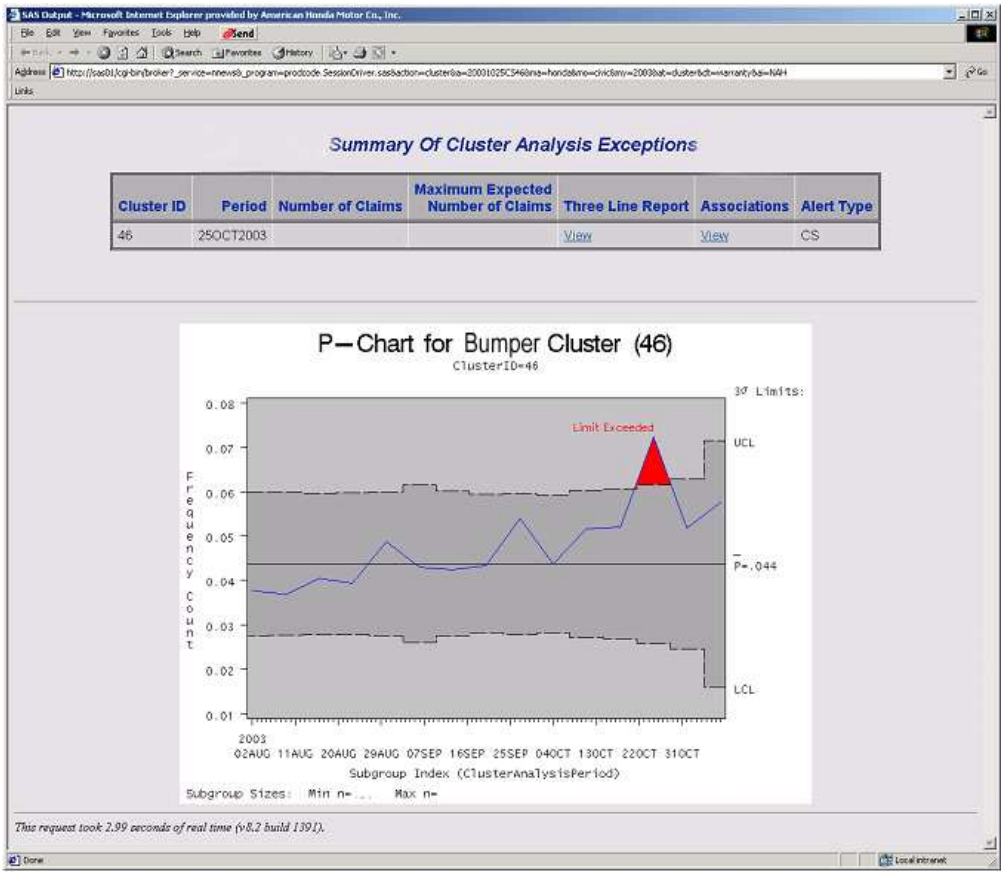

Figure 3.

CONCLUSION

The initial results of this methodology of deploying SAS Text Miner and SAS QC have been very positive. Systematic analysis has been enabled for text data that was too large to attempt to read. Working directly with the domain experts has increased both the usefulness of the models and their acceptance within the company. The process continues to evolve. Further automation has reduced the time required to bring a new clustering model into production and new ideas on increasing the homogeneity of clusters will be tested.

REFERENCES

SAS Institute Inc., *SAS/STAT Users Guide, Version 8, Volume 1*, Cary, NC: SAS Institute Inc., 1999.

SAS Institute Inc., Getting Started with SAS Text Miner, Cary, NC: SAS Institute Inc., 2002.

CONTACT INFORMATION

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

John Wallace, Principal
Business Researchers, Inc.
74 Mallorca Way
San Franicsco, CA 94123
Work Phone: 415-377-4759
Email: jwallace@businessresearchers.com
Web: http://www.businessresearchers.com