Paper 010-29

Using SAS® at SAS: The Mining of SAS Technical Support

Annette Sanders, SAS Institute Inc., Cary, NC
Craig DeVault, SAS Institute Inc., Cary, NC

ABSTRACT

Over the last decade, the amount of data that has been collected and stored has increased dramatically. Statistical methods have been, and continue to be, refined and perfected to yield valuable answers to business problems and goals using standard qualitative and quantitative data analysis methods. However, an overwhelming portion of all data collected is actually unstructured text. The evaluation of textual data is a relatively new focus area of statistics. Classical methods that generate answers to business problems can be enhanced effectively by coordinating the analysis of textual data with SAS® Enterprise Miner™ and SAS® Text Miner.

The Technical Support Division at SAS has not escaped this deluge of data. Although we collect data for every question or problem that we receive in order to manage the delivery of answers and solutions, we have never used the data as a tool for discovering underlying behavior or patterns of software and support issues -- until now.

INTRODUCTION

One of the first things you probably learn about SAS Institute (well, maybe after you hear about the M&Ms, the gym, the tennis courts, and putting green, etc.) is that when you have a question about our software, technical support consultants are easily accessible. SAS Technical Support provides customers with the resources they need to answer any questions or solve any problems they encounter when using SAS Software. With all of the different solutions and products that SAS offers, we have amassed a lot of data regarding customer inquiries and concerns.

SAS Technical Support uses an internally-developed application called SIRIUS (SAS Institute Resource for International User Support) to manage an inquiry. This tool provides an international interface to customer contact information, technical databases, and e-mail and fax tools. It also stores all customer contact notes, summaries of conversations with developers, and notes that consultants make during the resolution process.

Each time a customer contacts SAS Technical Support, a track is opened. Aside from the pages that contain notes, there are 89 variables in a track. These variables include general information such as name, company, site number, phone number, e-mail address, operating system, and the version of SAS used. Other variables pertain specifically to the question asked. The primary variables are tracking number, product, topic (for example, which procedure within the product is the area of concern), consultant, create date and time, last contact time, and resolve date. All of this information is stored in a data warehouse that we use to identify some general underlying trends. For example, we obtain reports on the number and percentage of tracks opened in 2003 in the Americas that were resolved within 24 hours. However, in most cases, the information collected provides only a high-level understanding; you must go to the notes to review the specifics of the customer's query. Because it is a daunting task to read through the minutiae of all tracks, only to come away with anything beyond a vague idea about some common issues, we needed to find an alternative to this approach. It is here that SAS Text Miner plays a vital role. SAS Text Miner is an add-on product that runs in the SAS Enterprise Miner environment. SAS Text Miner is designed specifically for the analysis of text.

FIRST THINGS FIRST

As with any analysis, you must first identify the business problems and concerns that you are trying to gain insight into. Just because you are involved in analyzing unstructured data does not mean that you should take an unstructured approach. We looked at our mission statement and gained additional input from management about our business problems and concerns to come up with the following goals:

- identify a set of terms to improve automatic classification of incoming electronic tracks in order to decrease the percentage of misrouted tracks
- identify specific areas that need better documentation or more examples on our Web site
- communicate the areas that need better documentation and Web examples to the Education Division so that they can incorporate these issues in SAS courses
- identify common concerns that can be addressed by frequently asked questions (FAQs)
- determine whether product concerns are related to usage or bugs.

A LOOK AT THE WORLD

From the beginning, we recognized that there were some limitations to what we were trying to accomplish. First, although SAS Text Miner can handle multiple languages[1], the document collection must contain the same language. Beyond that, while we have some experience with German and Latin, neither of us are well versed in other languages to be able to analyze a document collection and obtain meaningful results. Therefore, we needed to narrow down the number of documents in our collection to an amount that we, and SAS Text Miner, could handle. However, we wanted to be sure that we were not limiting our document collection to the extent that we could not draw inferences.

As previously mentioned, SIRIUS is a tool that handles tracks from all over the world. With SIRIUS we can identify the volume of tracks in all SAS offices.

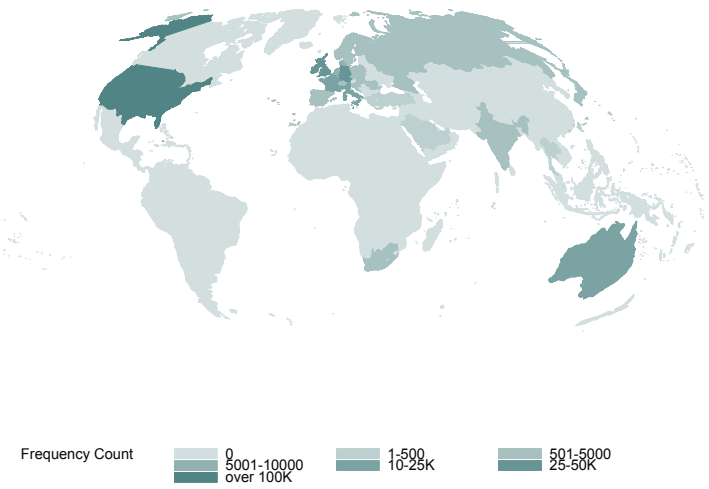


Figure 1. Track Volume during 2001-2003

As shown in Figure 1, the Technical Support Office in Cary, NC, SAS Headquarters, handled the highest number of tracks from 2001 to 2003, covering almost 400,000 tracks. In fact, this office worked on over 15 times the number of tracks than the office with the next highest volume of tracks. Naturally, this would be expected because most of SAS Research and Development (R&D) is located in Cary, NC, Additionally, because of the global differences in Time Zones, it makes the most sense for tracks to be handled in the same location as the research and development activities.

This eliminated any concern that the analysis of North American English-based tracks would not be representative of customer concerns.

THE CORPUS[2] SCOPE

Even when we limited our analysis to tracks from the United States, we were still left with a vast amount of data. We initially thought that we could use SAS Text Miner to obtain a set of terms that would help automate the routing of incoming electronic tracks. By parsing the terms from a customer e-mail, we hoped to use the scoring code from SAS Text Miner to assign the correct topic, subtopic, operating environment, etc., to each track.

We began this process by creating a SAS data set that contained the text from each track in a separate observation. Because the tracks in SIRIUS need to be retrieved from a collection of binary and text files, a SIRIUS application developer wrote a program that created a directory that contained individual text files for each track. Once we had this directory, it was easy to create the necessary SAS data set by using a macro provided with SAS Text Miner. This macro, `%tmfilter`, can read several different types of text data files, such as .txt, .doc, .html, .pdf, etc., and create a SAS data set from them. The macro for the SIRIUS track data call is

```
%tmfilter (dataset=sugi29.alltracks, dir=c:\texttracks)
```

where DATASET= specifies the name of the resulting data set, and DIR= specifies the location of the text files. The resulting data set includes two primary variables for use in the SAS Text Miner node: FILTERED, the variable that specifies the location of each of the document files; and TEXT, a portion of the text from the document files. SAS Text Miner only uses the TEXT variable in the results window so that you can get a general idea of each document without having to view the whole document. Document analysis is performed on the full document -- the document that is specified by the FILTERED variable.

For SAS Text Miner to evaluate this data, it must first parse the text in these documents into separate words or noun groups, perform one of two possible dimension reduction techniques or a combination of these techniques, and use the resulting information from the dimension reduction to cluster the documents. However, before you enable SAS Text Miner to perform a dimension reduction transformation, be sure that you are not diluting your analysis by including non-informative terms, and that your data is as clean and consistent as possible. These concerns are addressed with stop or start lists and synonym lists, respectively. Creating good start or stop lists is crucial in obtaining valid and useful results. These lists limit the terms that are included in the analysis. With a start list, you can specify which terms you want to include in your analysis; any other terms will not be included in the document by

[1] SAS Text Miner supports Dutch, English, French, German, Italian, Portuguese, and Spanish to varying degrees.

[2] A corpus is a collection of similar documents.

term matrix. A stop list has the opposite functionality; it is a list of terms that you want to remove from your analysis. Words that occur in all the documents, for example, a, an, the, be, are, is, with, by, etc., should be included in a stop list. Domain knowledge is essential for creating meaningful lists! The methods used to develop these lists are discussed extensively in the section entitled "Pre-Transformation Dimension Reduction".

It is impossible for us to have extensive knowledge about all areas of incoming tracks. SAS wouldn't have 180 consultants in the United States alone if this was the case. In addition, a stop list for SAS/STAT would be completely different from a stop list for the SAS/ACCESS Interface to Oracle applications. In fact, some words that are included in one stop list for one product might be very important and discriminating terms for another product. It would not be possible to have one stop list for all SAS products; it is unfeasible and would yield invalid results. At this point, we realized that evaluating all tracks in the United States was not a viable option. For purposes of our initial analysis, we could not use SAS Text Miner without significant input from consultants who have domain knowledge for each product that SAS develops.

Having domain knowledge of SAS Enterprise Miner, we decided to limit our collection of data to SAS Enterprise Miner tracks from the Cary office (North American tracks) that have a resolve date between January 1, 2001 and December 31, 2003. 3854 tracks occurred within this time period.

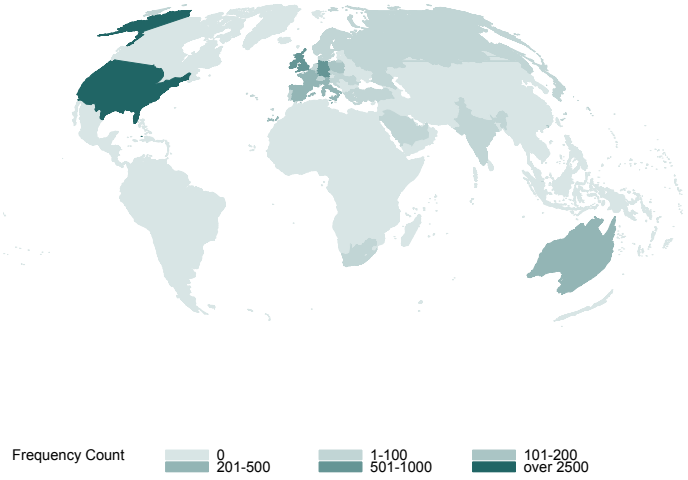


Figure 2. Enterprise Miner Track Volume during 2001-2003

We again wanted to evaluate whether our focus was too constrained and, therefore, produced another world map of SAS Enterprise Miner track volume for each global office. Aside from the obvious and expected frequency differences with all tracks, the pattern shown in Figure 2 is very similar to the one shown in Figure 1.

We concluded that we should perform the analysis on SAS Enterprise Miner tracks that originated in North America.

THE INITIAL ANALYSIS

Our goal is to use SAS Text Miner to enhance the information that we can obtain from structured data. In order to understand how this process can be enhanced, let's first take a look at some of the information that we already have. The first analysis looks for trends. Remember from our introduction that each track has a topic. In SAS Enterprise Miner, a topic is typically a node, which is a data mining tool, but it also has values such as "install/setup," and "client/server," "interface," "access," "model repository," etc. In all, there are almost 50 topics in SAS Enterprise Miner. For each of the top 15 most frequently used topics, we ran a Chi-Square test using the FREQ procedure in order to test for trends in the frequencies of these topics across the three years of data. The results of this analysis are shown in Table 1.

TOPIC	2001	2002	2003	p-value	Total
missing	381	325	353	0.1085	1059
tree	187	187	185	0.9929	559
install/setup	262	128	162	0.0001	552
other	272	131	78	0.0001	481
regression	115	71	109	0.0031	295
client/server	100	69	72	0.0263	241
text miner	0	30	130	0.0001	160
score	74	48	38	0.0015	160
neural	53	55	50	0.8867	158
assessment	59	42	54	0.2282	155
input data	43	31	45	0.2357	119
em cluster	54	19	27	0.0001	100
associations	41	33	16	0.0044	90
sas code	41	19	20	0.0031	80
transform	30	17	18	0.0893	65

Aside from the trends, Table 1 shows that the topic with the highest frequency count is "_missing_"; note also that "other" is the fourth most frequently occurring topic. This is one of the areas that we expected SAS Text Miner to shed some light on. We wanted to identify what types of customer concerns and questions are in these topic areas. These types of topic assignments are certainly a concern because we have no indication of what the track is about, and we are missing important frequency count information. This makes it very difficult to identify areas that have the most impact on our support resources.

Without more information about what is contained in these topic areas, you should cautiously consider any of the subsequent results.

Table 1. Chi-Square Test for Trends across Each Enterprise Miner Topic

For example, the "install/setup" topic has a "U-shaped" trend. This can be interpreted that SAS, Release 8.2, has been out in the marketplace for several years, and customers are not re-installing this release. The increase in 2003 is due to the installation of hot fixes and vertical add-on products. In addition to this, SAS Technical Support has noticed a pattern of hardware upgrades that generally occurs every couple of years. However, if any of the "install/setup" issues are misclassified by not assigning a topic or assigning a topic of "other", you can expect to see different results.

We saw another trend with the "text miner" topic. SAS Text Miner was not available to customers in 2001, but there was a definite upward trend in the number of tracks from 2002 to 2003. This confirms what we expected because as more customers use this software, the number of inquiries increases. There was also a consistent trend pertaining to the "tree" topic. Throughout the three years of data, the number of tree tracks is nearly uniform. The consistent nature of this trend indicates that the tree node, which is the most popular modeling method in SAS Enterprise Miner, has maintained a steady stream of use.

Rather than evaluating patterns within each topic, we wanted to look at all topics and years concurrently. The resulting matrix is 44 by 3. SAS Text Miner can handle matrices well beyond these dimensions. However, in this case, we used correspondence analysis to determine if we could reduce and explain these 44 topics and 3 years in one or two dimensions. We used the *%plotit* macro to obtain the correspondence analysis plot shown in Figure 3.

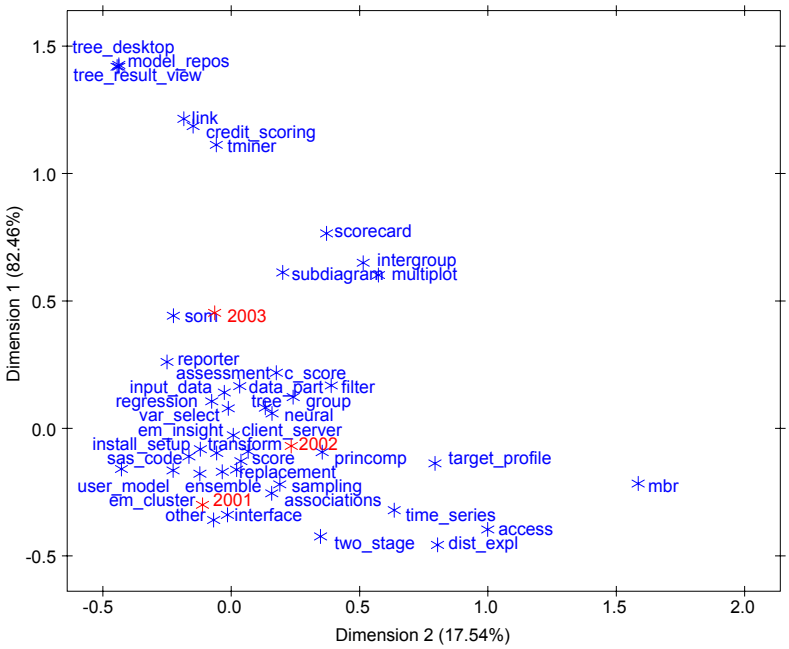


Figure 3. Correspondence Analysis of Enterprise Miner Topic by Year

Dimension 1 accounts for 82% of the total Pearson Chi-Square statistic. The one thing that is apparent from this plot is the probable meaning of Dimension 1. It seems to be a time dimension. Each year is higher in the Dimension 1 values than any previous year. Also, some of the topics that are at the top of Dimension 1 are more recent topics that were not available in all years. New modules such as the Tree Desktop Viewer, Credit Scoring, Text Miner, Tree Result Viewer, and the Model Repository are plotted near the top of Dimension 1. Some of the older topics that show descending trends, as shown in Table 1, have lower Dimension 1 values and are shown toward the bottom of the vertical axis. Some of these topics are "Insight," "Interface," "Access," "Distribution Explorer," and "Two-stage [model]."

The interpretation of Dimension 2 is something that is not immediately apparent. It accounts for the remaining 18% of the total Pearson Chi-Square statistic. Most topics are near the left side of the Dimension 2 axis. In considering the topics that had greater values for Dimension 2, we realized that this dimension appears to be inversely related to the relative frequency of each topic. The topics that correspond to higher values for Dimension 2 do not occur in our top 20 list in Table 1.

BACK TO THE CORPUS

SAS Text Miner identifies the number of occurrences of each term in each document. It then creates a document by term matrix where each column represents one of the distinct terms in the collection, and each row represents a document. In order to understand this matrix, SAS Text Miner performs a Singular Value Decomposition by default, which is one type of dimension reduction technique. Hopefully, this provides a picture of how the terms are related. A complete discussion of the mathematical foundation for the Singular Value Decomposition technique is provided by Albright[1].

Before we began the dimension reduction, we wanted to more thoroughly consider our corpus. When we started looking at our documents we noticed that another task, beyond effective stop and synonym lists, was necessary. For starters, some sections of each document are duplicated. Frequently, sections of the document contained an original e-mail in addition to a copy of the original e-mail when a response was received. This can cause extreme biases in our results; after all, the analysis is based on term frequency. We had to remove the duplicate information. Unfortunately, there was no algorithm that could perfectly identify these occurrences. When duplicated, the original e-mail was not uniquely tagged. In addition to this, occasionally, something that was marked as an original e-mail was actually the first time that it occurred in the document, for example, both the query to and the answer from R&D were included in the track. Therefore, we manually skimmed through each document to strip duplicate e-mails and make the data as clean as possible.

Another concern surfaced at this point. As we went through the documents, we noticed that each consultant in SAS Technical Support has slightly different work habits. Some consultants are concise and use shorthand in the problem descriptions, while other consultants are far more verbose. The differences in document length can create a bias. Also, the advanced features of SAS Text Miner, stemming, part-of-speech tagging, and entity identification, are designed for well-formed grammatically correct sentences [2]. Once again, because the basis of the SAS Text Miner analysis is word frequency, we recognized that we must carefully evaluate the approach requested of SAS Text Miner by considering the impact of each option. We discuss the choices we made in the section entitled "Back, Again, to the Corpus".

We now had the documents in a form ready to be addressed by SAS Text Miner. The next step was to parse all the terms in each document. In the initial consideration of your corpus, it is a good idea to only use the parsing feature of SAS Text Miner. There is no point in computing either of SAS Text Miner's dimension reduction techniques until you can identify the appropriate document by term matrix to pass to these tools.

There are several parsing features that you can use to help increase the richness of the terms data set based on your needs. In the Parse property settings of the node, you can select to identify, and, in some cases remove, terms that have the following characteristics:

- words that occur in a single document
- same word but a different part of speech
- stemmed words as root form
- entities
- numbers
- punctuation
- noun groups.

Using SAS Enterprise Miner 4.3 or 5.1, you can set up a simple flow of an Input Data Source node followed by the Text Miner node. Using the variables created by the *%tmfilter* macro (as described in the section entitled "The Corpus Scope"), the TEXT variable was set to a role of text and the FILTERED variable was set to a role of text location. We wanted to get an idea of how many words existed in our corpus, so we also set the Stem Terms, Different Parts of Speech, and Noun Groups options to "NO" and the Terms in a Single Document option to "YES." Because we were not yet interested in transforming the document by term matrix, we set the Compute SVD option to "NO." We set all other options to their defaults.

Using the exported terms data set, we summed the frequency of each term and found that our corpus contained a total of 3,992,766 words. The terms data set had 38,847 observations, which represented the number of unique words. For comparison, we found a copy of "War and Peace" by Leo Tolstoy [3] on the Internet, and using the Web crawler feature of the *%tmfilter* macro, we created a data set where each chapter was a document. "War and Peace" had 39,423 unique words and 553,147 total words. When comparing these two document collections, the number of unique words was very similar. We then configured SAS Text Miner to identify and remove words that occurred in a single document and used stemming to create a list of synonyms. The results are summarized in Table 2.

Corpus	# of Unique Terms	# Terms after Synonym & Stop List
Enterprise Miner tracks	38,847	32,764
"War and Peace"	39,423	10,205

"War and Peace" achieved a **74%** reduction in terms, whereas, the SAS Enterprise Miner collection only achieved a **16%** reduction.[3] This confirms that the SAS Enterprise Miner collection of tracks is very complex.

Table 2. SAS Enterprise Miner versus "War and Peace" Term Comparison

LOW INFORMATION TERMS

We now had a terms data set that was reduced by default stemming and removing terms that occurred in a single document. Using the transformation and clustering tools, we could now identify related sets of documents (clusters). SAS Text Miner reports a list of descriptive terms that are used to help identify cluster themes. Additional details about how these descriptive terms were identified are explained in the section entitled "Tweaking SAS Text Miner Options". In this initial foray into document clustering, we noticed that consultants, days of the week, and special characters were listed as descriptive terms. Table 3 shows the descriptive terms for clusters that were most likely modeling clusters.

[3] Remember that the documents frequently contain SAS code and system terminology.

Cluster1	Cluster3	Cluster5	Cluster6	Cluster10
+ regression	+ craig	+ value	+ variable	neural
+ target	+ split	+ target	+ set	+ regard
+ craig	stat	+ input	+ friday	+ estimate
+ score	+ tree	+ set	+ resolve	+ network
+ change	+ decision	+ code	+ craig	stat
+ value	em tree	+ node	+ node	+ annette
+ sample	+ annette	+ variable	stat	+ model
+ node	+ select	+ craig	=	+ value
+ model	+ value	+ score	+ wednesday	info
+ assessment	+ result	+ model	+ mike	+ answer
stat	+ node	+ annette	+ cluster	+ craig
+ tab	+ regard	stat	+ monday	+ variable
+ variable	+ resolve	data	+ size	+ result
+ set	+ set	+ change	<	+ request
+ chart	+ variable	+ select	+ thursday	+ help
+ resolve	+ one	+ create	+ update	+ subject
+ wednesday	+ wednesday	but	num	+ size
=	+ page	+ numb	sat	=
Data	+ model	+ run	+ answer	+ other
+ help	+ numb	+ regard	+ subject	+ node

The order is shown from highest to lowest probability of occurrence. Evaluating these terms can help you identify the nature of the cluster. However, descriptive terms are terms that are most likely to occur in a cluster; they do not represent terms that occur in all documents within that cluster. Also, there are only 20 terms presented here. The absence of a term does not mean that it is not associated with the cluster. This could result in erroneous conclusions. For example, in Table 3, "Tuesday" does not occur in the first 20 descriptive terms for any cluster, which suggests that you should not contact SAS Technical Support on a Tuesday if you have a modeling question. (We assure you that there are statisticians working in SAS Technical Support on Tuesdays!)

Clearly, it is necessary to further develop the stop and synonym list. It is important to drop any terms from the terms data set that are not meaningful to the goal of the analysis. There is no advantage in having irrelevant or uninformative terms in your data set; it only increases processing time due to the added dimensions of the document by term matrix.

Table 3. Descriptive Terms for Modeling Clusters

Additionally, it complicates the interpretation of your analysis and might contribute to invalid results.

For example, including meaningless terms can lead to improper classification. To illustrate this point, we used a more manageable document collection. We used quotes taken from the NC State University Wolfpack athletics Web site that recapped the 2003 football games and 2003-2004 basketball games [4]. Twenty-seven documents were used -- each quote was considered a separate document. We included a variable to indicate whether the quote pertained to football or basketball. We then read these documents into a SAS data set and set up a default flow in SAS Enterprise Miner.

We ran the Text Miner node using the default settings, and 5 clusters resulted. None of the initial clusters were entirely composed of either all football or all basketball quotes.

When evaluating the descriptive terms, one author of this paper (he shall remain nameless) considered the terms "need" and "with" uninformative and added them to the stop list (although, the other author would like to point out that when it comes to Wolfpack basketball, a strong argument could be made for including the term "need" in the analysis). We ran the Text Miner node again, excluding those terms, and the size and descriptive terms of the clusters changed. The resulting clusters became purer regarding the sports that they include. This simple example shows how sensitive the classification results can be when potentially trivial terms are added or removed from the stop list.

PRE-TRANSFORMATION DIMENSION REDUCTION

In order to obtain informative results from SAS Text Miner, you should consider several ways to reduce the dimensions of your document by term matrix before you begin the mathematical portion of the analysis. Start or stop lists and synonym lists are an integral part of this task. Recall that a start list enables you to specify which terms you want included in your analysis, and a stop list specifies which terms you want removed from your analysis. A synonym list collapses similar terms into a parent term so that they are included in the document by term matrix only as the parent value. The stemming feature of SAS Text Miner can greatly assist you in identifying a preliminary list of synonyms. Despite the extremely unstructured nature of our data, stemming was still a useful tool to begin reducing our dimensions. Also, *because* the data was so unstructured, using different parts of speech was pointless. This alone can result in a substantial reduction of dimensions. A word that is identified as both a noun and a verb will be represented by two columns in the document by term matrix. By not using different parts of speech, the term will be represented by one column.

Although time consuming, creating comprehensive start, stop, and synonym lists are essential steps for pre-transformation dimension reduction. For our analysis, the terms that should not be included were far more obvious

than the terms that must be included; therefore, we used a stop list rather than a start list. We did not want to spend time creating synonyms for terms that we would drop; therefore, for the first pass of this process, it was more efficient to create a stop list only. Also, although SAS Text Miner is well-suited to managing start, stop, and synonym lists at the same time, we found our thought process and typing to be more consistent when we began by only creating a stop list.

When you manage the stop, start, and synonym lists through the interactive SAS Text Miner interface, always make sure that you occasionally create a backup copy of these data sets. If anything interrupts the processing when saving either of these data sets, many days of work can be lost.

AUGMENTING THE STOP LIST

Regardless of how careful you are, when you work with a large number of terms, not all of them that should be dropped will be identified in your first pass. In fact, as you create the synonym list, you will encounter other terms that should be included in your stop list. We found that some tools were more helpful than others when considering the vast number of terms. We quickly realized that, although SAS Text Miner's link to documents is helpful, having domain knowledge usually made this step unnecessary for identifying the context of most terms.

You can easily identify terms that should not be included in your analysis by sorting your terms list in different ways. Different sort orders will produce groups of terms, and these groups will make it easier to identify many term issues. First, terms that occur in all or almost all the documents will not be useful for distinguishing between different types of documents. Sorting the data by the number of documents that each term occurs in is a quick way to eliminate a large number of these types of terms. Also, an alphabetical sort will find groups of similar terms that are not informative. When we sorted our list alphabetically, we found groups of many similar names, for example, Andersen, Anderson, Andre, Andrea, Andreas, Andrew, and Andy. (We did not want to automatically drop name entities because many statistical tests are identified by the originating name, and we definitely wanted to include these names in our analysis.) You might also find that sorting by weight and evaluating the lower-weighted terms will identify chunks of uninformative terms. Ultimately, however, even after you pull out specific groups, while the list of terms that you keep might be marginally smaller, you still need to consider uninformative terms from the perspective of corpus content. This is where domain knowledge is essential.

AUGMENTING THE SYNONYM LIST

When you are not using different parts of speech, the synonym list in SAS Text Miner is comprised of two variables: TERM and PARENT. The variable TERM contains the parsed term. The variable PARENT contains the value that TERM is mapped to. You might have noticed that the default synonym list available with SAS Text Miner contains only two terms, each associated with a different parent. However, as shown in Figure 4, the stemming process essentially created additional synonyms and reduced the number of terms from 6123 to 2244.

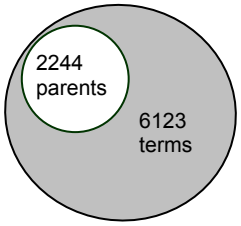


Figure 4. SAS Text Miner Term Reduction: 3879

The two main features that SAS Text Miner provides in the parsing and stemming process are tokenization and a dictionary-based stemmer. In the existence of grammatically correct and pristine English data, employing a dictionary stemmer can significantly reduce the number of terms. The stemmer in SAS Text Miner does more than truncate the term; it removes the suffix so that you get meaningful and correctly-spelled parent terms. We did obtain some reduction. However, our data included a lot of SAS code, operating system options and terminology; the dictionary-based stemmer does not address these terms.

Tokenization is the process of identifying terms. A term might not be just a single word but a group of words, for example, noun groups. We used the noun group option to identify and include noun groups in our document by term matrix.

There were many instances where the noun group was more informative than the separate terms themselves. For example, "neural network" is more informative than the two separate terms "neural" and "network," primarily because the second term can mean something entirely different when it is not paired with "neural." The same word can have different meanings depending on how it is used in the document; this is known as polysemy. However, we also found that the dictionary-based stemmer often left noun groups as separate terms; for example, "CHAID analysis" and "CHAID tree" were not assigned to the same parent.

Using noun groups will increase the dimension of your document by term matrix because each individual term in the noun group is still in that matrix, therefore, the terms themselves will also occur separately. SAS Text Miner does not make an exception when dealing with the individual terms even when they exist in a noun group. The terms are counted as a group and also as individual terms, which can potentially introduce some bias. However, the single

terms cannot be removed from the matrix either because they need to be counted when they occur outside the noun group. Otherwise, another bias will result. Ideally, when the noun group occurs, it is most appropriate to count only the group and not the individual terms. This is currently not the way that SAS Text Miner is designed to handle this issue, but it is under consideration for a future release.

SAS Text Miner is designed to create parents from terms that exist in your collection of documents. Sometimes, however, a more simplified or better term than any that exists in your document collection can be most appropriate. Consider the occurrence of "chi-sq," "chisq," and "chi^2 value"; the best choice for the parent would be "chi-square." If it doesn't occur, then it cannot be assigned through the interactive SAS Text Miner GUI. Perhaps, as an extension to this case, you want your base term to represent a more robust set of terms -- terms that really are synonyms in the usual sense. For example, "improve," "enhance", and "ameliorate" are handled with the same parent. In each of these cases, manual intervention is required.

Finally, SAS Text Miner spends a lot of time processing while adding and displaying equivalent terms. Therefore, to reduce the amount of time that is spent creating synonyms, and to further automate the synonym process with different algorithms, we used the Analysis tool in dfPower Studio[4] to assist us in developing our synonym list. dfPower Studio analysis identifies inconsistencies and improves data accuracy and integrity. Other than a very easy–to-use interface, dfPower Studio also provides a mechanism for identifying similar terms that is different from SAS Text Miner.

AUGMENTING THE SYNONYM LIST WITH dfPOWER STUDIO

dfPower Studio is limited to text fields of 250 characters, but you will have already parsed your document with SAS Text Miner. Now, you need to extract the terms data set produced by SAS Text Miner to use as input to dfPower Studio. dfPower Studio employs a process that involves finding several permutations of data that are similar, and then assigning the most frequently occurring permutation in that group as the Standard [5]. (A Standard in dfPower Studio is comparable to a parent in SAS Text Miner.) However, dfPower Studio does not accept summarized data as input. Therefore, you need to restructure the terms data set from SAS Text Miner by expanding the number of observations of each term to the frequency in which it occurs. After you create this data set, it can be read directly into dfPower Studio by using an ODBC driver. When dfPower Studio evaluates this data set, it summarizes the number of occurrences of each term. Naturally, this brings you back to where you started with the terms data set from SAS Text Miner, but because the terms are already parsed the process takes a minimal amount of time.

After you have your data in dfPower Studio, use the "Text(Scheme Build)" match definition phrase analysis tool and specify a value for sensitivity in order to build an ordered table of terms. After you build the table of terms, known as a report, you will perform a Scheme Build. The sensitivity option indicates how similar each term should be in order to be included in the same cluster. As dfPower Studio evaluates each term, it assigns a match code. The sensitivity value controls how close the match codes must be in order to group terms together. The value can range from 50 to 95, with a default of 85. The higher the value of sensitivity, the closer the match code must be for each term. The match definition is used to generate a sorted table of similar terms and is subsequently used when assigning a standard to a group of terms. At this point, dfPower Studio uses a "Smart Clustering" match definition algorithm to group what it considers to be similar data. For example, "all," "*all*," and ">>all" are presented next to each other in the resulting table. Without dfPower Studio's "Smart Clustering," a strictly alphabetical sort will not place these items together. "Smart Clustering" makes it easier to apply the same standard to inconsistent terms. The logic used to identify similar terms when using the "Text(Scheme Build)" match definition involves several techniques. The most significant of these, for purposes of our analysis, are regular expression normalization, standardization, phonetic routines, and sensitivity. Regular expression normalization corrects misspellings and transforms terms that have embedded characters, for example, it removes a dash between telephone numbers. Standardization reduces the term by extracting the initial portion of the string. For example, we saw that "1st Question," "Initial Question," and "Final Question" were all grouped together. As shown in Figure 5, the dfPower Studio process reduced the number of terms from 9847 to 3181.

[4] dfPower Studio is developed by DataFlux®, a wholly owned subsidiary of SAS Institute Inc.

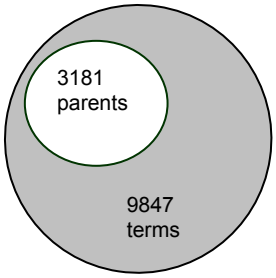


Figure 5. dfPower Studio Term Reduction: 6666

It is important to note that although the algorithms used in dfPower Studio can significantly reduce the manual effort required in building an effective synonym list, *the resulting standardization is only a starting point*. The results should be carefully reviewed. The dfPower Studio GUI makes it easy to evaluate the Standards that it builds. The dfPower Base – Analysis Editor window displays the report (list of terms and their frequency) on the left side and the Standardization Scheme (the list of terms and their chosen standard) on the right. Once again, different sort orders will reveal standards that need to be adjusted. You will also be able to specify a standard that does not occur as a term in the report window. Another useful feature enables you to directly compare your report to your scheme. dfPower Studio identifies terms that do not have a standard in the color red, which makes it easy to identify terms that need a standard that was not automatically applied.

Although the dfPower Studio GUI is very easy to use, one disadvantage is that you are not able to directly add to your stop list as you are going through your list of terms. We found that having a simultaneous invocation of SAS Text Miner and dfPower Studio was the best approach so that we could immediately add terms to our stop list as they were identified.

After you complete your review and adjust the standards as needed, it is easy to get the dfPower Studio data back into a format that can be used by SAS Text Miner. dfPower Studio can save the data to a comma delimited text file. You can easily get this into a SAS data set by using the SAS Import Wizard. Finally, perform a simple DATA step to reformat the data as needed by SAS Text Miner. Remember from the section entitled “Augmenting the Synonym List” that the synonym list in SAS Text Miner is comprised of two variables (when you are not using different parts of speech): TERM and PARENT. Using the code generated from the Import Wizard and adjusting it to meet the format required by SAS Text Miner resulted in the following SAS DATA step:

```
data DFCLEAN.fromdfpower;
   %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
   infile 'C:\My Documents\datacleanse\eminer scheme.txt'
           MISSOVER DSD lrecl=32767 ;
   informat term $136. parent $136.;
   format term  $136. parent $136.;
   input term parent;
   /* set ERROR detection macro variable */
   if _ERROR_ then call symputx('_EFIERR_',1);
run;
```

After we merged the dfPower Studio schemes with the SAS Text Miner synonyms, we saw some interesting results, which are shown in Figure 6. For the most part, the power of each tool was unique; the two tools had applied a synonym to the same term only 2919 times. However, each tool also seemed to have a common approach to some synonyms. Out of the 2919 times that both tools came up with a synonym, they agreed with what that synonym should be 2352 times – approximately 80% of the time! Also, as shown in Figure 7, the number of terms in our final synonym list was reduced from 13,051 to 4188.

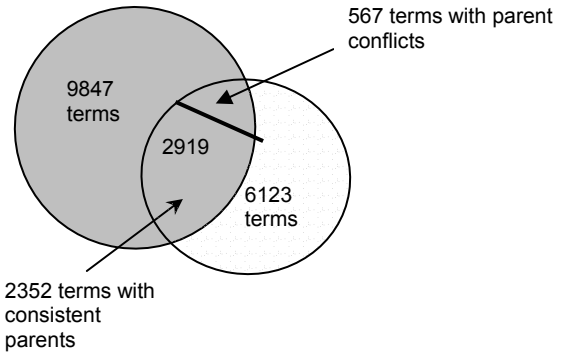


Figure 6. dfPower Studio Merged with SAS Text Miner Synonyms

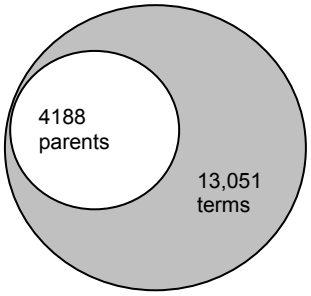


Figure 7. Final Synonym Data Set

BACK, AGAIN, TO THE CORPUS

When the synonym and stop lists were in a satisfactory state, we ran SAS Text Miner again using these lists and the default settings. This resulted in 11 clusters. The descriptive terms for a subset of clusters are shown in Table 4.

Obs	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
RMS Std	0.074	0.082	0.081	0.077	0.069	0.062
# docs	319	209	274	157	356	181
1	+ tree	+ license	+ documentation	+ estimate	+ connect	+ stat
2	+ stat	+ fail	+ procedure	+ network	+ server	+ validation
3	+ split	+ session	+ proc	+ neural network	+ libname	+ model
4	+ decision tree	+ start	+ association	+ weight	+ input data	+ partition
5	+ train	+ log	+ analysis	+ model	+ project	+ lift chart
6	+ choose	+ connect	+ cluster	+ train	+ create	+ assessment
7	+ save	+ setinit	+ stat	+ case	+ client/server	+ assess
8	+ sample	+ unix	+ response	+ stat	+ open	+ lift
9	+ value	+ error	+ size	+ value	+ access	+ regression
10	+ model	+ server	+ hope	+ regression	+ library	+ manager
11	+ target	+ client	+ result	+ tab	+ file	+ chart
12	+ change	+ client/server	+ numb	+ more	+ data	+ show
13	+ observation	+ project	+ node	+ node	+ data set	+ data set
14	+ select	+ system	+ answer	+ information	+ code	+ node
15	+ input	+ site	+ unknown	+ result	+ up	+ tab
16	+ tab	+ file	+ information	+ answer	+ start	+ result
17	+ node	+ install	+ value	+ hope	+ make	+ select
18	+ variable	+ now	+ know	+ variable	+ need to	+ look
19	Into	+ send	+ question	+ one	+ user	+ change
20	+ option	+ problem	+ only	+ data	+ should	+ value

Table 4. SAS Enterprise Miner Clusters and Their Descriptive Terms

The descriptive terms for each cluster show us what we expect from grouping SAS Enterprise Miner tracks. Cluster 1 contains decision tree tracks. Clusters 2 and 5 have client/server and installation tracks, but the difference between them is not immediately apparent from these results. Cluster 3 contains association and clustering tracks. Cluster 4 is a neural network cluster. Cluster 6 has both assessment and regression tracks. These clusters have good separation of the terms in our document collection. The Root Mean Squared Standard Deviation (RMS Std) values are low for all of the clusters. The RMS Std value is a distance measure that indicates how tightly packed documents

are within a cluster. The distance between a document and a cluster is the Mahalanobis distance, sqrt((x-u)'S(x-u)), where u is the cluster mean, and S is the inverse of the cluster covariance matrix [6].

The results are very close to what we anticipated based on a simple pareto chart. (See Figure 8.) The number of documents in each cluster, other than Cluster 5, is very consistent with the frequency distribution of topics.

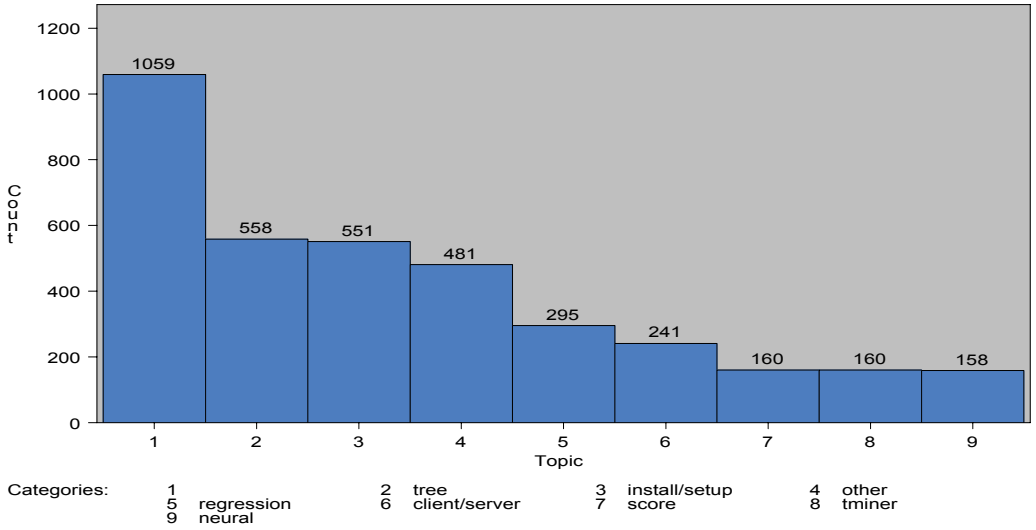


Figure 8. Pareto Chart Accounting for 80% of the Highest Frequency Counts

As shown in Figure 8, the most frequent topics for SAS Enterprise Miner are “Missing”, “Decision Trees,” “Installation and Setup,” “Other,” “Regression,” and “Client/Server.” These six topics account for 67% of the SAS Enterprise Miner tracks. Two of these topics, “Other” and instances where no topic was specified, are meaningless. These types of topic assignments might occur when a customer sends an e-mail that contains questions from many different topic areas, or the consultant might not have assigned a topic to the track. Let us evaluate the distribution of these two topics over the 11 clusters that SAS Text Miner identified to determine whether or not these categories can be collapsed in subsequent analyses. Table 5 shows a cross-tabulation of the “Missing” and “Other” topics with the cluster ID from SAS Text Miner.

Topic	**Cluster ID**										
Frequency **Cell Chi-Square**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
	11 0.0286	61 0.151	22 0.0432	9 0.0586	117 0.2057	14 0.7285	73 0.3244	163 1.4464	113 0.9433	199 0.2564	41 0.8387
Other	4 0.0658	31 0.3472	11 0.0993	5 0.1347	44 0.473	2 1.6748	25 0.7458	94 3.3251	35 2.1686	97 0.5895	10 1.9282
Total	15	92	33	14	161	16	98	257	148	296	51

Table 5. Cross-Tabulation of Uninformative Topics by Cluster ID

The patterns in cluster assignment seem very similar, and a Chi-Square test to determine whether there is an association between topic and cluster was only marginally significant. The Chi-Square results are shown in Table 6.

Statistic	**DF**	**Value**	**Prob**
Chi-Square	10	16.5771	0.0843

Table 6. Chi-Square Test for Association between Topics and Clusters

As shown in Table 5, Cluster 8 has the highest Cell Chi-Square value in both rows.

Cluster8
+ information
+ message
+ send
+ site
+ unknown
+ other
+ include
+ need
+ version
+ question
+ request
+ find
+ receive
+ system
+ install
+ size
+ answer
+ know
+ note
+ response

Table 7 shows a list of the top 20 descriptive terms for Cluster 8. Unfortunately, this list of descriptive terms is not very descriptive in identifying what this cluster has in common. It is also unfortunate that it has the second highest frequency. It might be that the only thing that it has in common is that it doesn't have anything in common; that is to say, this is a cluster of outliers. It might be useful to request terms that are more descriptive when we run SAS Text Miner, or alternatively, we might want to subset our document collection to include only those documents assigned to Cluster 8, and run SAS Text Miner on this collection. However, further consideration of Cluster 8 will be left for another discussion.

Without Cluster 8, as you can see from Table 8, there is no evidence to indicate that a topic of "Missing" is different from a topic of "Other."

Statistic	DF	Value	Prob
Chi-Square	9	10.8461	0.2864

Table 8. Chi-Square Test for Association between Topics and Clusters, without Cluster 8

Table 7. Descriptive Terms for Cluster 8

TWEAKING SAS TEXT MINER OPTIONS

Up to this point, we simply accepted most defaults chosen by the Text Miner node. Now by using a topic variable as a target variable, we can identify how well the various options created clusters that most purely predict the topic area. However, rather than evaluating every single permutation of options, we decided to narrow our focus by first considering the expected effect of each option as it relates to our corpus and to our business goals. (Note that the choice of options might be entirely different for your document collection and goals.)

As we consider each of the options that are available in the properties dialog of SAS Text Miner, recall from the section entitled "Pre-Transformation Dimension Reduction " that we used the following parse options during the data cleansing process: stem terms, remove terms that occur in a single document, and include noun groups. We must also consider the transformation and clustering properties.

TRANSFORM PROPERTIES

There are two types of transformations in this property:

- singular value decomposition (SVD)
- roll-up terms.

SVD dimension, resolution, and scale

The SVD option creates orthogonal columns that characterize the terms data set in fewer dimensions than the document by term matrix. A high number of SVD dimensions usually summarizes the data better but requires a lot of computing resources. In addition, the higher the number, the higher the risk of fitting to noise. The default transform method is the SVD. In most cases, including the runs on the SAS Enterprise Miner tracks data, the SVD had consistently better results.

The default number of dimensions that the SVD creates is 100. In an option related to the SVD transformation, you can select low, medium, or high resolution for the number of dimensions. The resolution determines the number of computed dimensions set by the maximum SVD dimension property that should be used by the clustering algorithm. Low, medium, and high resolutions correspond to 2/3, 5/6, and 6/6 (100%) of the computed dimensions, respectively. When you begin your analysis you will probably want to use the low resolution so that you can reduce the computing resources required by the clustering algorithm, but you can still evaluate additional SVD dimensions in order to determine whether further dimensions are needed for clustering. After you have determined the adequate number of SVD dimensions for clustering, there is no need to compute more SVDs than you are going to use.

You can evaluate further SVD dimensions by using a scree plot. The scree plot is found in the Interactive Results and shows the proportional amount of variance explained by each additional SVD dimension. When this plot starts to flatten out, this is a sign that the information contained in the later dimensions does not add much to the model. With the SAS Enterprise Miner corpus, the percent of misclassification rate differed by only 0.9 whether the number of SVD dimensions was set to 25, 50, or 100. When combining the misclassification rate with the scree plot, we decided to compute and use 25 SVD dimensions. SVD dimensions beyond 25 do not account for a noticeable increase in incremental variance, and we are probably just fitting to noise.

Another option related to the SVD transformation is whether to scale the SVDs. Scaling creates SVDs with equal variance. You want to consider scaling your SVD dimensions if you have rare targets and find a lot of variance accounted for in later dimensions. With the scree plot, you see a pattern that is decreasing very slowly. We did not have rare targets and found large proportions of variance described in the first several SVD dimensions; therefore, we did not select the scaling property.

Term and Frequency Weights

There are two types of weight schemes that we can apply to the terms: global frequency weighting schemes and local frequency weighting schemes.

The three local frequency weighting schemes that are found in the frequency weighting property are:

- Binary
- Log
- None

When you choose the Binary weighting scheme, a term is given a weight of 1 when it is found in a document and 0 if it is not. The Log weighting scheme takes the log of the number of times a term is found in the document. The None weighting scheme simply uses the frequency itself. In our analysis, Log is the only local weighting scheme that makes sense to use. A term could be in a track and yet have very little to do with the purpose of the track. If we used a Binary weighting scheme, then the term would be given the same weight as the main topic of the track. Also, we do not want to unduly weight a frequently used term, such as “log”, that might be relevant but is more likely to appear more frequently than its importance in defining that relevance.

The eight types of global frequency weighting schemes that are found in the term weight property are:

- Entropy
- Inverse Document Frequency
- Global Frequency*Inverse Document Frequency
- Normal
- None
- Chi-Squared
- Mutual Information
- Information Gain

The last three methods use a target variable. We did not consider the last three, because we were trying to identify concepts beyond a topic area of SIRIUS in our final analysis, in other words, clusters that do not have a target. However, in the initial approach to identify the appropriate settings, we used the SIRIUS topic as a target. We ran the remaining term-weighting schemes through all topic areas and followed this by a Memory-Based Reasoning node as described by Cox and Albright [7]. The normal global term weight resulted in the highest misclassification rate, and the Entropy weighting scheme gave us the lowest misclassification rate.

Roll-Up Transformation

The next section of the Transform property is the method of roll-up terms. This method can be used with or without the SVD method. The roll-up terms method weights the terms that are in each document and then uses the terms with the top N highest weights, where N is identified in the subsequent “No. of Rolled-Up terms” property. If you use roll-up terms for clustering, then your documents (observations) are more strongly correlated than if you use SVD without scaling. When you scale the SVD by singular values, SVD results in completely uncorrelated observations. However, because of that dependence, your clusters will be more compact. This method does not take into consideration the entire term list for clustering, so if you are short on computational resources, this might help.

Clustering with the Roll-Up method works best when you have short documents that have very little word overlap. Also, the more thorough you are in creating your synonym list, the better you will do with roll-up terms. If you want to run both the SVD and Roll-Up methods together, you should also select the “Drop Other Terms” property. The Text Miner node will run the Roll-Up method first and then take the terms that make the cutoff and run the SVD transformation on these remaining terms. In evaluating our corpus, we did not want to limit the analysis to the top

weighted terms because the weight values decreased very slowly. Therefore, we needed a large number of roll-up terms to account for even moderate weights.

CLUSTER PROPERTIES

Cluster is the last property group of the Text Miner node and two clustering methods are available: expectation-maximization and hierarchical clustering. The expectation maximization clustering technique is similar to k-means clustering, but without the postulation of similar cluster shapes and sizes. The hierarchical clustering technique uses the CLUSTER procedure in SAS/STAT [8] with Ward's minimum-variance method [9]; it employs a non-recursive method. For our analysis, there was no need for the hierarchical method because there is no restriction on the sub-cluster being related to the parent cluster. Therefore, we used the expectation maximization clustering technique exclusively. Another reason we used this technique is because any outliers that arise from the clustering are placed in one outlier cluster. This cluster contains all outlier documents. This is accomplished by setting the "Ignore Outliers" property to the default "NO".

In order to help determine the theme of the cluster, SAS Text Miner reports descriptive terms. You can set the number of descriptive terms, m, that you want SAS Text Miner to report in the "Descriptive Terms" property. Remember that descriptive terms are created by evaluating term frequency, whereas the dimension reduction techniques use a transformation of the document by term frequency matrix. SAS Text Miner computes a binomial probability for each 2*m terms, where m is the number of descriptive terms requested; from these, the top m descriptive terms that are shown have the highest binomial probabilities. Not only are they the top m descriptive terms, but the terms are presented in order of descending probability. Because we didn't expect terms to occur infrequently within each document, we increased the value of descriptive terms from the default of 5, up to as many as 40, in order to gain a more thorough understanding of the cluster theme.

Descriptive terms might not always help discriminate between the clusters; in fact, we had instances where terms were equally likely to occur in multiple clusters. Remember, clusters are determined from the SVD dimensions, not directly from frequencies. In these cases, evaluate which SVD dimension is most important for each cluster and which terms have the highest and lowest SVD weights for that dimension.

BEYOND THE PROPERTIES OPTIONS

By not categorizing tracks that we know are different and then running an analysis on those tracks separately, we were causing some problems. In other words, (as shown in the section entitled "Back, Again, to the Corpus") we did not want to dilute our analysis and spend processing time finding dimensions that were already identified with the SIRIUS topic. We could do this by using the topic variable, which will reduce the SVD space for each topic area and should help us find more subtle differences within each topic.

THE "FINAL" ANALYSIS

After all of this preliminary effort, you might be thinking "Where's the Beef?" As you consider this question now, did it prompt any thoughts about what we might be thinking -- perhaps something that will not be in the written word but that can be read "between the lines"? Additionally, what we are really wondering about at this point is how we can possibly fit the rest of this topic into a 20-page limit when we are already on page 15 and have just now gotten to the explanation of the results. It is exactly these kinds of concepts, although hopefully more relevant and important to the topic of the paper, that regardless of how carefully you choose the options in SAS Text Miner, and regardless of how perfect they are for your corpus, will never be uncovered. SAS Text Miner can only uncover relationships of concepts that are stated in the document collection.

Fortunately, there were several interesting relationships in the subsequent analysis. We will present a few here. You can probably read between the lines and realize that this is still a work in progress.

DECISION TREE TRACKS

When the options identified in the section entitled "Tweaking SAS Text Miner Options" were applied to each individual topic area, one of the more useful relationships we found involved the decision tree tracks. Refer to Tables 9 and 10 for details regarding these clusters.

LABEL	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7
RMS Std	0.1412	0.1241	0.118	0.05651	0.1421	0.1218	0.1204
Freq	56.0000	30.0000	207.000	8.00000	50.0000	56.0000	69.0000

Table 9. RMS Std and Frequency for Decision Tree Clusters

Table 9 shows that 207 tracks (43%) are clustered in Cluster 3. Also, the RMS Std value indicates that the clusters are all fairly dense. The RMS Std for Cluster 4 is small. Generally, the larger the cluster, the larger the RMS Std value -- you expect variance to increase when you have more documents in the cluster. Therefore, a large cluster

with a relatively low RMS Std indicates a strong cluster. When you have a large cluster with a relatively large value for RMS Std, you should consider further processing to find sub-clusters. Conversely, a small cluster with a large RMS Std means that it is most likely an outlier cluster.

Looking at Table 10, it is easy to determine the nature of each cluster by looking at only the top 25 descriptive terms. We define the clusters as follows:

1. Performing decision tree analysis using the underlying procedure code
2. Model assessment, possibly involving a profit matrix or priors as specified through the target profiler
3. Using the interactive interface to produce a decision tree
4. Advanced statistics and alternative models
5. CHAID analysis
6. Errors that occur when performing decision tree analysis
7. Printing issues/Tree Results Viewer/Tree Desktop Application

Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7
+ proc split	+ assessment	+ train	+ weight	+ merge	+ log	+ printer
+ procedure	+ model assess	+ variable	noise variance	+ search	+ font	+ tree data set
Proc	+ profit	+ score	occurrence	+ branch	+ fail	+ install
+ project	+ profit matrix	+ model	inversely	chaid	+ error	+ file>print
+ second	+ cost	+ rule	+ least-squares	+ categorical	+ character	+ print tree
+ error	+ classify	+ decision tree	assumption	+ compute	+ parameter	+ download
+ log	advanced tab	Interactive	estimation	+ assign	+ traceback	+ device
+ time	+ target profile	+ stat	mse	Gini	+ happen	+ txt file
+ line	+ event	+ save	approximation	+ measure	+ fix	+ print
+ occur	+ proportion	+ node	effective sample size	+ evaluate	+ large	setup
+ cause	+ subtree	+ change	frequencies	+ approx	+ problem	+ emf
+ start	+ decision	+ group	related statistics	Maximum	+ memory	+ directory
+ step	+ partition	+ leaf	approx. estimates	+ basic	+ message	+ viewer
+ diagram	+ validation	+ show	samp. probability	+ method	+ send	desktop
+ system	+ default	Owner	frequency values	+ advance	+ line	+ export
+ process	+ correct	+ split	correct results	+ few	+ receive	+ description
+ need	+ target	+ make	error variance	+ apply	+ occur	+ tree diagram
+ code	+ understand	+ all	weighted l-s	+ leaf	+ small	+ require
+ open	+ criterion	+ tab	usual assumption	+ great	+ flow	+ output
+ appear	+ model	+ value	weighted estimation	+ statistician	original	+ box
+ site	+ win	+ automatic	significance tests	+ split	+ pass	+ file
+ point	+ advance	Into	standard errors	+ value	+ file	+ application
+ file	+ choose	+ result	actual frequencies	+ obs	+ contain	+ system
+ send	Yes	+ sample	neural network weights	+ large	+ code	+ new
+ contain	Sun	+ data	regression coefficients	+ choose	should	+ type

Table 10. Top 25 Descriptive Terms for the Decision Tree Clusters

As previously mentioned, a large percentage of tracks seem to be related to interactive training. SAS R&D was ahead of the game, and a new interface was recently developed to accommodate a richer set of features. This indicates that SAS should give more attention to this topic in courses, FAQs, and supporting documentation.

REGRESSION TRACKS

Table 11 shows the RMS Std and the number of documents for each cluster found in the subset of Regression tracks. SAS Text Miner also found a cluster very similar to the one found in the Decision Tree subset. Notice in Table 12 that Cluster 1 includes some of the same statistical terms and, as in the case of Cluster 4 for the Decision Tree subset, this cluster is very small with only 6 documents.

LABEL	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
RMS Std	0.07816	0.1386	0.1535	0.1239	0.1357	0.1215
Freq	6.00000	32.0000	29.0000	82.0000	35.0000	83.0000

Table 11. RMS Std and Frequency for Regression Clusters

Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
+ weight	+ parameter est	+ occur	+ prior	+ interaction	Memory
+ integer	+ estimate	+ happen	+ lift chart	+ documentation	+ log
correct results	+ deviation	+ pass	+ model	+ title	+ error
error variance	reference level	+ cause	+ training data	+ include	out of
noise variance	+ effect	+ rerun	+ assessment	+ request	+ file
weighted least-squares	+ ratio	+ return	+ validation	+ incorporate	+ wait
usual assumption	+ constrain	+ diagram	+ select	+ answer	+ receive
weighted estimation	+ ascend	+ problem	+ probability	+ information	+ talk
then standard errors	+ level	+ replicate	+ target	+ url	+ observation
actual frequencies	+ parameter	+ attribute	+ create	+ status	+ data set
neural network weights	+ intercept	+ error	+ sample	+ proc	+ process
other analyses	+ average	+ delete	+ tree	+ term	+ attach
noise-variance weights	+ odd	+ open	+ response	+ unknown	+ miss
explicit support	+ compute	+ thing	+ base	+ logistic	+ send
fractional part	+ category	back	+ stat	+ score	+ note
nonnegative integers	+ class var	+ log	+ note	+ procedure	+ user
weight variable	+ interpret	+ make	+ know	Sun	+ flow
+ approximate	+ last	+ need	+ case	dmreg	Back
significance tests	standard	+ program	+ numb	+ type	+ problem
weighted analyses	+ difference	+ project	+ regression	+ provide	Into
+ truncate	glm	+ send	+ train	+ result	+ train
Estimation	+ categorical	again	+ full-screen	em reg subtopic	+ time
+ represent	+ code	+ stepwise	+ node	+ most	Like
+ handle	Analysis	+ close	+ logistic	+ difference	+ no
frequency variable	+ class	+ list	+ observation	+ stat	dmreg

Table 12. Top 25 Descriptive Terms for the Regression Clusters

Further review of Table 12 enables us to easily identify, with one exception, the nature of these clusters:

1. Advanced statistics and alternative models
2. Creation and interpretation of class variable parameterization
3. Errors that occur when performing regression analysis
4. Model assessment, possibly involving priors; potentially compared to a tree model
5. (Performing regression analysis using the underlying procedure code?)
6. Computer resource issues

The one exception was Cluster 5. However, SAS Text Miner makes it easy to subset the results in this cluster and review the original documents. Because we have only 35 documents in this cluster, reviewing these documents is manageable. Reviewing Cluster 5 documents uncovered the following theme:

5. Performing regression tree analysis using the underlying procedure code and/or interpretation of interactions.

Part of the reason that the nature of Cluster 5 was not apparent is because it might actually contain two, similar but separate, sub-clusters. Rather than using the non-recursive approach of hierarchical clustering, we processed only these 35 documents through Expectation-Maximization clustering again. Indeed, SAS Text Miner did find 2 clusters, and only 2 clusters. They were clearly identified through their descriptive terms as the same two concepts that we determined from manually reviewing the documents.

TRACKS WITH "OTHER" OR "MISSING" TOPIC

We anticipated that we would obtain a large number of clusters for the tracks that were not assigned to an informative topic value. The reason is that we believed that the content of these tracks was either not related to any of the 44 topics or was too complex to assign a value to. Curiously, SAS Text Miner reported only 6 clusters.

The RMS Std value and frequency for each cluster is found in Table 13.

LABEL	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
RMS Std	0.112	0.108	0.157	0.103	0.107	0.105
Freq	176.000	188.000	109.000	206.000	162.000	340.000

Table 13. RMS Std and Frequency for Missing or Other Clusters

Each cluster has a relatively large number of documents that are reasonably dense. The descriptive terms are shown in Table 14. Once again, the descriptive terms are helpful in identifying the nature of each cluster, but this instance required work beyond just looking at the descriptive terms. In cases where we easily determined a concept, we classified them with parenthesis and question marks as follows:

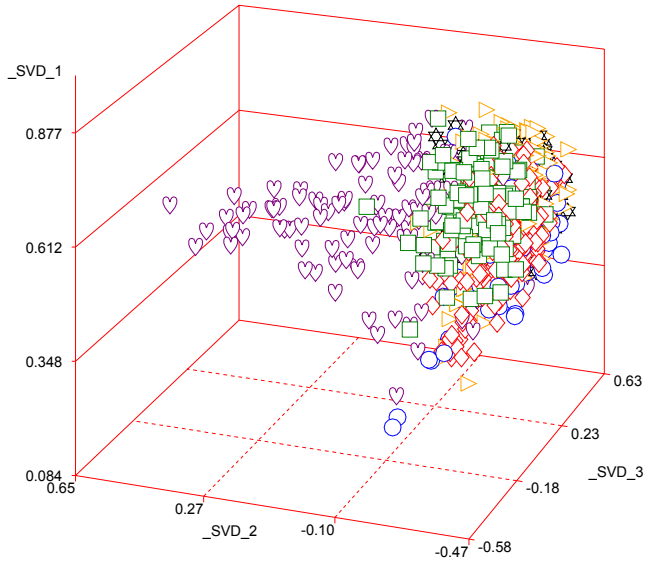
1. General feature availability
2. Statistical question regarding multiple modeling nodes
3. (Computational Resources?)
4. Installation/setinit issues
5. Client/Server issues
6. (Errors/Bugs/Locked diagrams?)

Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
+ provide	+ stat	Real	+ need	+ remote	+ lock
+ web site	+ score	+ tmp	+ license	+ library	+ diagram
+ doc	+ target	+ line	+ expire	+ assign	+ file
+ link	+ input	+ cpu	+ pc	+ unix	+ hot fix
+ documentation	+ model	+ character	+ setinit	+ connect	+ open
+ data mining	+ node	+ return	+ install	+ libname	+ delete
+ web	+ variable	real time	+ thin client	+ connection	+ copy
+ mine	+ regression	+ second	+ service	+ client/server	+ fix
+ tool	+ data set	+ observation	+ client	+ server	+ project
+ analytical	+ value	+ complete	+ server	+ define	+ close
unknown	+ tab	+ statement	+ installation	+ process	+ happen
+ development	+ data	+ log	+ contract	+ project	+ problem
+ information	+ select	+ pass	+ contact	+ start	+ error
+ question	+ code	+ occur	+ base	+ directory	+ window
+ score	+ mean	+ execute	+ site	+ access	+ full-screen
+ other	+ tree	+ error	+ connect	+ client	+ report
+ pm	+ case	+ end	+ software	+ create	Again
+ stat	+ default	+ entry	+ window	+ pc	+ save
+ model	+ sample	+ procedure	+ correct	+ transfer	+ click
+ application	+ result	+ cause	Both	+ automatically	+ occur

Table 14. Top 25 Descriptive Terms for Missing or Other Clusters

Both Clusters 3 and 6 required some further attention.

When reviewing some basic scatter plots involving the SVD dimensions, noting that Cluster 3 is identified by a heart symbol, you can see in Figure 9 that there is a lot of separation of Cluster 3 from the other clusters in the second SVD dimension.



In order to consider what was driving that dimension, we took a look at the list of Cluster 3 terms and evaluated the largest weights for the second SVD dimension. The largest weights are shown in Table 15. Note: you might also want to look at the lowest weighted terms to determine what is not important to that dimension.

What is immediately apparent is the number of terms with an equal (=) sign after the term. These are exactly the kinds of terms that we see in tracks when we are trying to get more in-depth information (from SAS logs) about errors and computer resources.

Figure 9. Plot of 1st Three SVD Dimensions for 3rd Missing or Other Cluster

_SVD_2	TERM	_SVD_2	TERM	_SVD_2	TERM
0.74602	errcode =	0.71772	uplist =	0.71001	string =
0.73375	vlist =	0.71678	prefix =	0.70913	privates =
0.72450	+ inds	0.71641	testds	0.70849	dslist =
0.72171	temp_l	0.71586	centry =	0.70758	catlist =
0.72097	newlist =	0.71510	role =	0.70601	open =
0.72070	datalib =	0.71442	level =	0.70512	dsinfo =
0.71935	desc =	0.71382	varlist =	0.70512	tmplist =
0.71868	+ vname	0.71133	flag =	0.70480	+ sampds
0.71794	dsl	0.71118	expsds	0.70478	name =
0.71794	savemeta =	0.71063	newmeta =	0.70156	Scrds
0.71772	dsl =	0.71010	meta =	0.70094	nobs =

Table 15. Highest SVD Dimension 2 Weights for 3rd Missing or Other Cluster

There were 4 sub-clusters found in Cluster 6 and their RMS Std and frequency values are shown in Table 16. Cluster 4 is a very small cluster that does not appear to be very densely packed. This is most likely an outlier cluster.

LABEL	Cluster1	Cluster2	Cluster3	Cluster4
RMS Std	0.125	0.151	0.0823	0.1654
Freq	102.000	197.000	29.0000	12.0000

Table 16. RMS Std and Frequency Counts for Sub-Clusters of Missing or Other Cluster 6

The cluster theme was often easily identified by the descriptive terms. In fact, as we suspected, there was a bug cluster theme as identified by the descriptive terms for Cluster 1 shown in Table 17. Cluster 3 is well described as a locked diagram theme. Further evaluation was needed for Cluster 2. In this case, we considered using a similar investigation as described in the previous sub-cluster analysis. However, we will not address that at this time.

Cluster1	Cluster2	Cluster3	Cluster4
memory	+ print	.lck extension	+ !sasroot\core
+ log	within	program halt	sas system help
+ cause	+ need	then search	+ accept
+ hot fix	+ directory	same filename	+ web
+ read	+ good	step instructions	+ language
+ apply	+ copy	wildcard	pack
+ node	+ install	wildcard search	+ default
+ happen	+ create	easy way	+ error
+ problem	+ select	+ explore	Html
+ corrupt	+ include	+ shoot	+ field
+ like	+ need to	approach	Microsoft
+ issue	+ make	toolbar	General
+ user	+ out	userid	+ address
+ time	+ back	+ halt	+ assign
+ send	more	+ circumvent	+ request
down	into	following message	+ encounter
+ process	+ change	+ tip	+ edit
+ rerun	only	online help	+ behavior
+ error	+ question	+ extension	+ recent
+ close	+ appear	+ lck file	Explorer
+ server	+ contain	+ lock	Never
+ occur	+ result	+ click	Might
Could	able	+ unlock	+ follow
+ information	same	+ search	+ appear
+ know	+ check	+ section	+ load
First	+ option	+ start	+ service
+ data	+ no	+ crash	+ download
+ request	+ save	+ instruction	+ display
Up	+ data set	+ project	+ server
+ size	+ look	+ attempt	Unknown

Table 17. Descriptive Terms for Sub-Clusters of Missing or Other Cluster 6

The one thing that remains consistent, whether we dig further or not, is that the choices for topic assignment in SIRIUS are not complete enough to identify the nature of these clusters. Based on these results, we must modify the list in order to obtain more meaningful information from fields directly available in each track.

CONCLUSION

Sometimes it is important to take a step back and look at the big picture. Usually when you do this, you eventually find yourself immersed in a whole new level of detail. In SAS Technical Support, we are often in the midst of parts and pieces. In writing this paper, we wanted to take a step back and, using SAS software, look at the big picture. Without fail we got that, and we also got much more – an interesting perspective.

We think that it is important to reiterate that while SAS Text Miner is designed for instances in which “sentences should be structured properly for best results, including correct grammar, punctuation, and capitalization” [2] our corpus was anything but that. Additionally, as a direct result from this analysis, we can respond to business goals specific to our corpus scope. Given the results, we also recognize that it will be worth the time to implement this very useful tool, SAS Text Miner, in order to address these goals in subsequent SAS Enterprise Miner topic areas and other technology/solution areas.

We have identified a large subset of tracks that relate to interactively training a decision tree. Especially now that we have a new tool to accomplish this, we need to be sure that its use and features are thoroughly documented, and that

the SAS Education Division includes complete coverage of this topic in their decision tree course. Similarly, education materials and further documentation should be developed for the Tree Desktop Application and the parameterization and interpretation of class variables. In each of these cases, some amount of coverage exists; however, we need to ensure that the specific questions that customers routinely bring to SAS Technical Support are adequately covered.

This leads us into model assessment involving multiple modeling tools, priors and profits; the Education Division has excellent coverage of this topic in "Predictive Modeling Using Enterprise Miner Software" [10]. Naturally, just because topic coverage is included in a course doesn't mean customers have attended the course. We should provide a link to this course from related FAQs on the SAS Technical Support Web site. We should also carefully consider the specific questions that come into Technical Support, and because SAS Text Miner has identified a group of tracks related to this topic it will be much easier to evaluate.

The nature of the aforementioned cluster themes applied to our business goals is broad topic coverage. In other instances, the cluster nature was much more specific and includes some areas where SAS Technical Support should incorporate FAQs, if not thorough technical documents. These areas include general CHAID approximation summary, general feature availability information, and computational resources issues.

In addition to these results, we recognize that we can make relatively minor adjustments to our SIRIUS topic values and gain a large amount of knowledge with very little effort, leaving us time to identify and resolve other areas of concern by using SAS Text Miner.

Indirectly, we will be better able to anticipate our customer's concerns and needs regarding SAS Text Miner. We were able to identify many usability concerns and, as in any software, bugs. So, the big picture for us is a path to a better product, better documentation, and better support for our customers.

REFERENCES

[1] Albright, Russell. "Taming Text with the SVD." January 7, 2004: 72 paragraphs. Available www.sas.com/apps/whitepapers/whitepaper.jsp?code=SDM5.

[2] "Introduction to Text Miner." In "SAS Text Miner Help." SAS OnlineDoc 9.1. 2003. SAS Institute Inc., Cary, NC.

[3] Haynes, Tony. Recaps of 2003 North Carolina State University Football and 2003-2004 North Carolina State University Basketball Games. 2004. Available gopack.ocsn.com/sports/m-footbl/recaps and gopack.ocsn.com/sports/m-baskbl/recaps.

[4] Tolstoy, Leo. "War and Peace". 1911. 365 Chapters. Available www.friends-partners.org/oldfriends/literature/war_and_peace/war-peace_intro.html.

[5] Data Flux. 2003. dfPowerStudio User's Guide.

[6] "SAS Text Miner Appendixes." In "SAS Text Miner Help." SAS OnlineDoc 9.1. 2003. SAS Institute Inc., Cary, NC.

[7] Albright, Russell, Cox, James and Daly, Kevin. 2001."Skinning the Cat: Comparing Alternative Text Mining Algorithms for Categorization." Proceedings of the 2nd Data Mining Conference of DiaMondSUG, Chicago, IL. DM Paper 113.

[8] SAS Institute Inc. 1999. *SAS/STAT User's Guide, Version 8, Volume 1*, Cary, NC: SAS Institute Inc.

[9] Ward, J.H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236 -244.

[10] SAS Institute Inc. 2003. "*Predictive Modeling Using Enterprise Miner Course Notes"*, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

We want to sincerely thank several people for their technical advice and encouragement. Specifically, thank you Russell Albright, James Cox, Lou Soleo, and Leslie Warren.

TRADEMARKS

CONTACT INFORMATION

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

Annette Sanders
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27587
Email: Annette.Sanders@SAS.com

Craig DeVault
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27587
Email: Craig.DeVault@SAS.com