Getting Started with
SAS® Text Miner 4.1

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2009. *Getting Started with SAS® Text Miner 4.1*. Cary, NC: SAS Institute Inc.

Getting Started with SAS® Text Miner 4.1

ISBN 978-1-59994-999-4

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, September 2009

1st printing, September 2009

Contents

Recommended Reading

- *Many of the concepts and topics that are discussed in additional product documentation for SAS Text Miner 4.1 (http://support.sas.com/documentation/onlinedoc/txtminer) and SAS Enterprise Miner 6.1 (http://support.sas.com/documentation/onlinedoc/miner) might also help you use SAS Text Miner 4.1.*

For a complete list of SAS publications, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative at:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: 1-800-727-3228
Fax: 1-919-531-9439
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Chapter 1
Introduction to Text Mining and SAS Text Miner 4.1

What Is Text Mining?

Text mining uncovers the underlying themes or concepts that are contained in large document collections. Text mining applications have two phases: exploring the textual data for its content and then using discovered information to improve the existing processes. Both are important and can be referred to as descriptive mining and predictive mining.

Descriptive mining involves discovering the themes and concepts that exist in a textual collection. For example, many companies collect customers' comments from sources that include the Web, e-mail, and call centers. Mining the textual comments includes providing detailed information about the terms, phrases, and other entities in the textual collection, clustering the documents into meaningful groups, and reporting the concepts that are discovered in the clusters. Results from descriptive mining enable you to better understand the textual collection.

Predictive mining involves classifying the documents into categories and using the information that is implicit in the text for decision making. For example, you might want to identify the customers who ask standard questions so that they receive an automated answer. Additionally, you might want to predict whether a customer is likely to buy again, or even if you should spend more effort in keeping him or her as a customer.

Predictive modeling involves examining past data to predict future results. Consider that you have a customer data set that contains information about past buying behaviors, along with customer comments. You could build a predictive model that can be used to score new customers: to analyze new customers based on the data from past customers. For example, if you are a researcher for a pharmaceutical company, you know that hand-coding adverse reactions from doctors' reports in a clinical study is a laborious, error-prone job. Instead, you could create a model by using all your historical textual data, noting which doctors' reports correspond to which adverse reactions. When the model is constructed, processing the textual data can be done automatically by scoring new records that come in. You would just have to examine the "hard-to-classify" examples, and let the computer handle the rest.

Both of these aspects of text mining share some of the same requirements. Namely, textual documents that human beings can easily understand must first be represented in a form that can be mined by the software. The raw documents need processing before the patterns and relationships that they contain can be discovered. Although the human mind comprehends chapters, paragraphs, and sentences, computers require structured (quantitative or qualitative) data. As a result, an unstructured document must be converted into a structured form before it can be mined.

What Is SAS Text Miner 4.1?

SAS Text Miner 4.1 is a plug-in for the SAS Enterprise Miner 6.1 environment. SAS Enterprise Miner provides a rich set of data mining tools that facilitate the prediction aspect of text mining. The integration of SAS Text Miner within SAS Enterprise Miner combines textual data with traditional data mining variables. A Text Miner node can be embedded into a SAS Enterprise Miner process flow diagram. SAS Text Miner supports various sources of textual data: local text files, text as observations in SAS data sets or external databases, and files on the Web. The Text Miner node encompasses the parsing and exploration aspects of text mining and prepares data for predictive mining and further exploration using other SAS Enterprise Miner nodes. The Text Miner node enables you to analyze structured text information, and combine the structured output of a Text Miner node with other structured data as desired.

The Text Miner node is highly customizable and enables you to choose among a variety of parsing options. It is possible to parse documents for detailed information about the terms, phrases, and other entities in the collection. You can also cluster documents into meaningful groups and report concepts that you discover in the clusters. You can use the Text Miner node in an environment that enables you to interact with the collection. Sorting, searching, filtering (subsetting), and finding similar terms or documents all enhance the exploration process.

The Text Miner node's extensive parsing capabilities include the following:

- stemming
- automatic recognition of multi-word terms
- normalization of various entities such as dates, currencies, percentages, and years
- part-of-speech tagging
- extraction of entities such as organizations, products, Social Security numbers, time, titles, and more
- support for synonyms
- language-specific analysis for English, German, Chinese, French, Spanish, Italian, and Portuguese

SAS Text Miner also enables you to use a SAS macro that is called %TMFILTER. This macro accomplishes a text preprocessing step and enables SAS data sets to be created from documents that reside in your file system or on Web pages. These documents can exist in a number of proprietary formats.

SAS Text Miner is a very flexible tool that can solve a variety of problems. Here are some examples of tasks that can be accomplished using SAS Text Miner:

- filtering e-mail
- grouping documents by topic into predefined categories

- routing news items
- clustering analysis of research papers in a database
- clustering analysis of survey data
- clustering analysis of customer complaints and comments
- predicting stock market prices from business news announcements
- predicting customer satisfaction from customer comments
- predicting costs, based on call center logs

The Text Mining Process

Whether you intend to use textual data for descriptive purposes, predictive purposes, or both, the same processing steps take place, as shown in the following table:

Action	Result	Tool
File preprocessing	Creates a single SAS data set from your document collection. The SAS data set is used as input for the Text Miner node and might contain the actual text or paths to the actual text.	%TMFILTER macro — a SAS macro for extracting text from documents and creating a predefined SAS data set with a text variable
Text parsing	Decomposes textual data and generates a quantitative representation suitable for data mining purposes.	Text Miner node
Transformation (dimension reduction)	Transforms the quantitative representation into a compact and informative format.	Text Miner node
Document analysis	Performs clustering, classification, prediction, or concept linking of the document collection.	Text Miner node or SAS Enterprise Miner predictive modeling nodes

Finally, the rules for clustering or predictions can be used to score a new collection of documents at any time.

You might not need to include all of these steps in your analysis, and it might be necessary to try a different combination of text-parsing options before you are satisfied with the results.

Accessibility Features of SAS Text Miner 4.1

SAS Text Miner includes accessibility and compatibility features that improve usability of the product for users with disabilities. These features are related to accessibility standards for electronic information technology adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended. SAS Text Miner supports Section 508 standards except as noted in the following table.

Section 508 Accessibility Criterion	Support Status	Explanation
When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.	Supported with exceptions.	The software supports keyboard equivalents for all user actions with the exceptions noted below: The keyboard equivalent for exposing the system menu is not the Windows standard Alt + spacebar. The system menu can be exposed using the following shortcut keys: (1) Primary window — Shift + F10 + spacebar, or (2) Secondary window — Shift + F10 + down key. The Explore action in the data source pop-up menu cannot be invoked directly from the keyboard, but there is an alternative way to invoke the data source explorer using the **View** —> **Explorer** menu.
Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.	Supported with exception.	Node run or failure indication relies on color, but there is also a corresponding pop-up message in a dialog box that indicates node success or failure.

If you have questions or concerns about the accessibility of SAS products, send e-mail to accessibility@sas.com.

Chapter 2
Learning by Example: Using SAS Text Miner 4.1

About the Scenario in This Book

This book describes an extended example that is intended to familiarize you with the many features of SAS Text Miner. Each topic in this book builds on the previous topic, so you must work through the chapters in sequence. Several key components of the SAS Text Miner process flow diagram are covered. In this step-by-step example, you learn to do basic tasks in SAS Text Miner, such as how to create a project and build a process flow diagram. In your diagram, you perform tasks such as accessing data, preparing the data, building multiple predictive models using text variables, and comparing the models. The extended example in this book is designed to be used in conjunction with SAS Text Miner software.

The Vaccine Adverse Event Reporting System (VAERS) data is publicly available from the U.S. Department of Health and Human Services (HHS). Anyone can download this data in comma-separated value (CSV) format from **http://vaers.hhs.gov**. There are separate CSV files for every year since the U.S. started collecting the data in 1990. This data is collected from anybody, but most reports come from vaccine manufacturers (42%) and health care providers (30%). Providers are required to report any contraindicated events for a vaccine or any very serious complications. In the context of a vaccine, a contraindication event would be a condition or a factor that increases the risk of using the vaccine. Please see the "Guide to Interpreting Case Report Information Obtained from the Vaccine Adverse Event Reporting System (VAERS)" available from HHS (`http://vaers.hhs.gov/info.htm`).

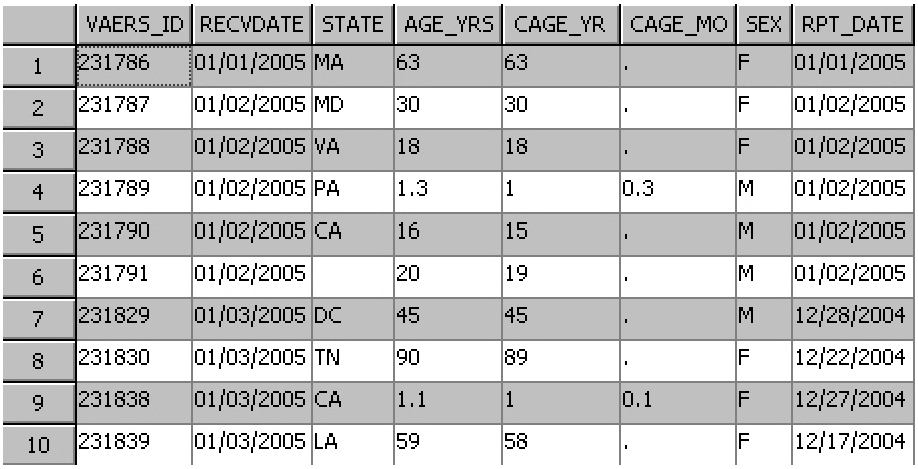
See the following in the **Getting Started Examples** zip file:

- ReportableEventsTable.pdf for a complete list of reportable events for each vaccine
- VAERS README file for a data dictionary and list of abbreviations used

Note: See "Prerequisites for This Scenario" on page 8 for information about where to download the **Getting Started Examples** zip file.

The following figure shows the first 8 columns in the first 10 rows in the table of VAERS data for 2005. Included is a unique identifier, the state of residence, and the recipient's age. Additional columns (not in the following figure) include an unstructured text string

SYMPTOM_TEXT that contains the reported problem, specific symptoms, and a symptom counter.



	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE
1	231786	01/01/2005	MA	63	63	.	F	01/01/2005
2	231787	01/02/2005	MD	30	30	.	F	01/02/2005
3	231788	01/02/2005	VA	18	18	.	F	01/02/2005
4	231789	01/02/2005	PA	1.3	1	0.3	M	01/02/2005
5	231790	01/02/2005	CA	16	15	.	M	01/02/2005
6	231791	01/02/2005		20	19	.	M	01/02/2005
7	231829	01/03/2005	DC	45	45	.	M	12/28/2004
8	231830	01/03/2005	TN	90	89	.	F	12/22/2004
9	231838	01/03/2005	CA	1.1	1	0.1	F	12/27/2004
10	231839	01/03/2005	LA	59	58	.	F	12/17/2004

In analyzing adverse reactions to medications, both in clinical trials and in post-release monitoring of reactions, keyword or word-spotting techniques combined with a thesaurus are most often used to characterize the symptoms. The Coding Symbols for Thesaurus of Adverse Reaction Terms (COSTART) has traditionally been the categorization technique of choice, but it has been largely replaced by the Medical Dictionary for Regulatory Affairs (MedDRA). COSTART is a term developed by the U.S. Food and Drug Administration (FDA) for the coding, filing, and retrieving of post-marketing adverse reports. It provides a keyword-spotting technique that deals with the variations in terms used by those who submit adverse event reports to the FDA.

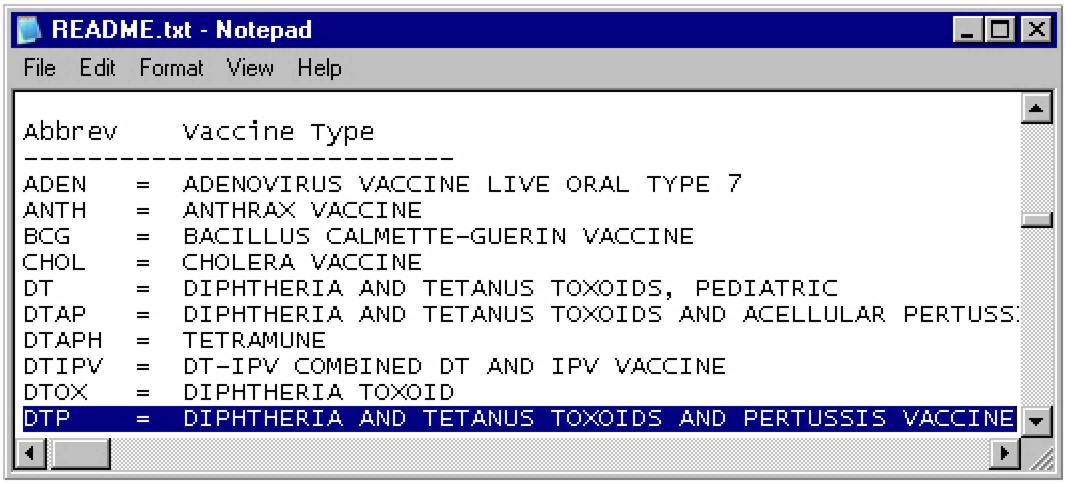
In the case of vaccinations, the COSTART system has been used. The FDA has used a program to extract COSTART categories from the SYMPTOM_TEXT column. Here are some of the variables used by the program:

- SYMPTOM_TEXT — reported symptom text
- SYM01- SYM20 — extracted COSTART categories
- SYM_CNT — number of SYM fields that are populated for a particular vaccination
- VAERS_ID — VAERS identification number

If you open the VAERS data for 2005 you can see that VAERS_ID **`231844`** has SYMPTOM_TEXT of **`101 fever, stiff neck, cold`** — the program has automatically extracted the COSTART terms that appear in column SYM01 to column SYM20 in the data file.

The VAERS table contains other columns, including a variety of flags that indicate the seriousness of the event (life-threatening illness, emergency room or doctor visit, hospitalized, disability, recovered), the number of days after the vaccine that the event occurred, how many different vaccinations were given, and a list of codes (VAX1-VAX8) for each of the shots given. There are also columns indicating where the shots were given, who funded them, what medications the patient was taking, and so on.

The README file taken from the VAERS Web site decodes the vaccine abbreviations. Note that some vaccinations contain multiple vaccines (for example, DTP contains diphtheria, tetanus, and pertussis). Here is a portion of the README file:

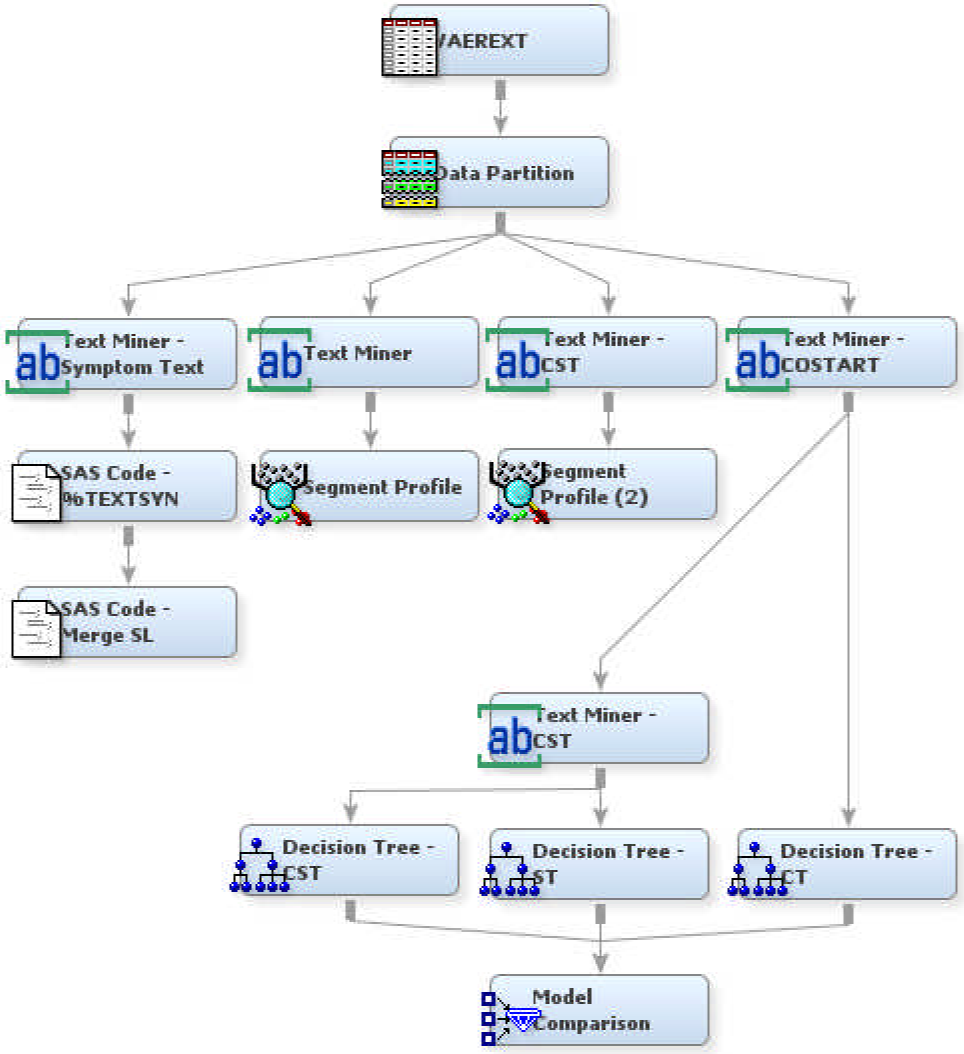


```
Abbrev    Vaccine Type
---------------------------
ADEN   =  ADENOVIRUS VACCINE LIVE ORAL TYPE 7
ANTH   =  ANTHRAX VACCINE
BCG    =  BACILLUS CALMETTE-GUERIN VACCINE
CHOL   =  CHOLERA VACCINE
DT     =  DIPHTHERIA AND TETANUS TOXOIDS, PEDIATRIC
DTAP   =  DIPHTHERIA AND TETANUS TOXOIDS AND ACELLULAR PERTUSSI
DTAPH  =  TETRAMUNE
DTIPV  =  DT-IPV COMBINED DT AND IPV VACCINE
DTOX   =  DIPHTHERIA TOXOID
DTP    =  DIPHTHERIA AND TETANUS TOXOIDS AND PERTUSSIS VACCINE
```

As you go through this example, imagine you are a researcher trying to discover what information is contained within this data set and how you can use it to better understand the adverse reactions that children and adults are experiencing from their vaccination shots. These adverse reactions might be caused by one or more of the vaccinations they are given, or they might be induced by an improper procedure from the administering lab (for example, a non-sanitized needle). Some of them will be totally unrelated. For example, perhaps someone happened to get a cold just after receiving a flu vaccine and reported it. You might want to investigate serious reactions that required a hospital stay or caused a lifetime disability or death, and find answers to the following questions:

- What are some categories of reactions that people are experiencing?
- How do these relate to the vaccination that was given, the age of the recipient, the place they received the vaccine, or other pertinent information?
- What factors influence whether a reaction becomes serious?
- How well are these factors captured by the automatically extracted COSTART terms?
- Is there any important information contained in the adverse reaction text that is not represented by the COSTART terms?

When you are finished with this example, your process flow diagram should resemble the one shown here:



Prerequisites for This Scenario

Before you can perform the tasks in this book, administrators at your site must have installed and configured all necessary components of SAS Text Miner 4.1. You must also perform the following:

1. Download the Getting Started Examples zip file under the SAS Text Miner 4.1 heading from the following URL:

   `http://support.sas.com/documentation/onlinedoc/txtminer`

2. Unzip this file into any folder in your file system.
3. Create a folder called Vaersdata on your C:\ drive.
4. Copy the following files into **C:\Vaersdata**:
   - Vaerext.sas7bdat
   - Vaer_abbrev.sas7bdat
   - Engdict.sas7bdat

Note: The preceding list of files might or might not be capitalized depending on the environment you are viewing them in.

How to Get Help for SAS Text Miner 4.1

Select **Help** ▸ **Contents** from the main SAS Enterprise Miner menu bar to get help for SAS Text Miner.

Chapter 3
Setting Up Your Project

About the Tasks That You Will Perform

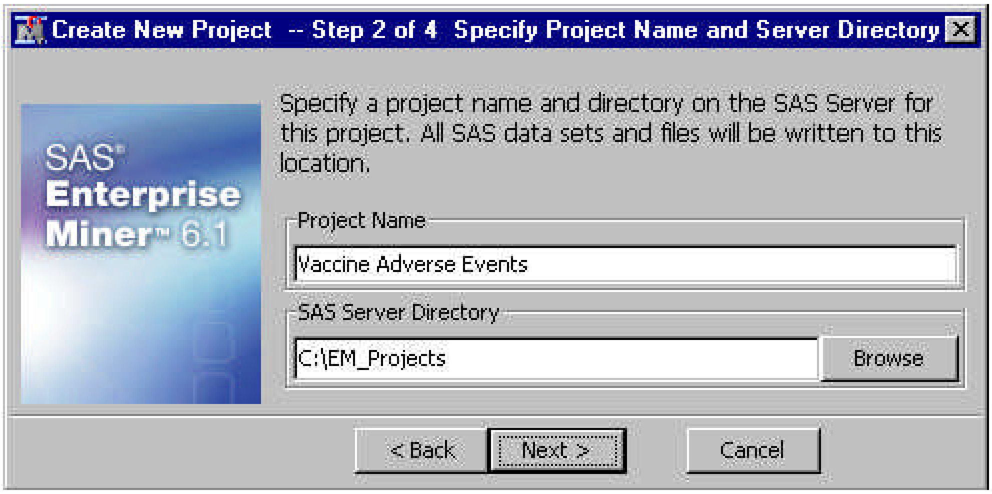
To set up your project, perform the following main tasks:

1. Create a new project where you will store all your work.
2. Define the VAERS data as a SAS Enterprise Miner data source.
3. Create a new process flow diagram in your project.

Create a Project

To create a project:

1. Open SAS Enterprise Miner.
2. Click **New Project** in the SAS Enterprise Miner window. The Select SAS Server page opens.
3. Click **Next**. The Specify Project Name and Server Directory page opens.



Create New Project -- Step 2 of 4 Specify Project Name and Server Directory

Specify a project name and directory on the SAS Server for this project. All SAS data sets and files will be written to this location.

Project Name: Vaccine Adverse Events

SAS Server Directory: C:\EM_Projects

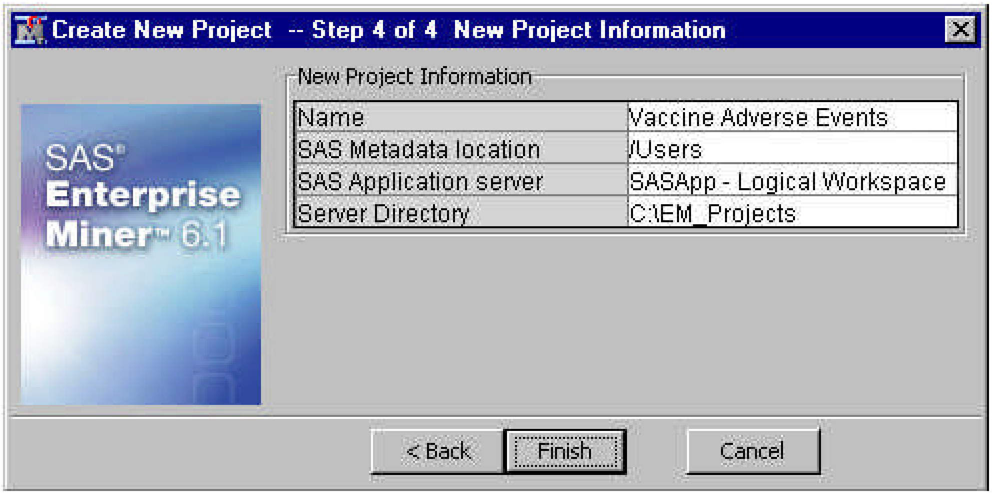
4. Type a name for the project, such as *Vaccine Adverse Events*, in the Project Name box.

5. In the SAS Server Directory box, type the path to the location on the server where you want to store data for your project. Alternatively, browse to a folder to use for your project.

   Note: The project path depends on whether you are running SAS Enterprise Miner as a complete client on your local machine or as a client/server application. If you are running SAS Enterprise Miner as a compete client, your local machine acts as its own server. Your SAS Enterprise Miner projects are stored on your local machine in a location that you specify, such as **C:\EM_Projects**. If you are running SAS Enterprise Miner as a client/server application, all projects are stored on the SAS Enterprise Miner server. If you see a default path in the SAS Server Directory box, you can accept the default project path, or you can specify your own project path. This example uses **C:\EM_Projects**.

6. Click **Next**. The Register the Project page opens.

7. Click **Next**. The New Project Information page opens.



Create New Project -- Step 4 of 4 New Project Information

New Project Information

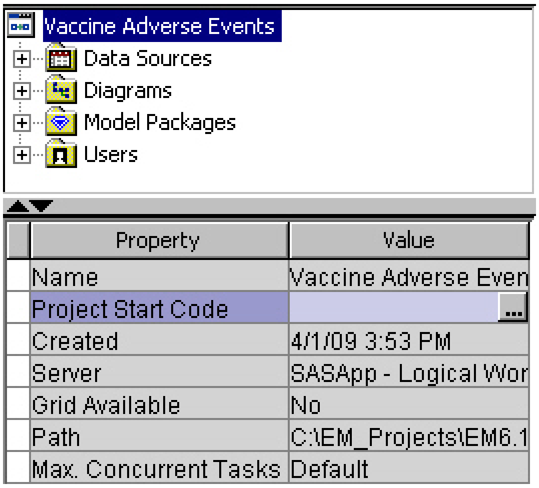
Name	Vaccine Adverse Events
SAS Metadata location	/Users
SAS Application server	SASApp - Logical Workspace
Server Directory	C:\EM_Projects

8. Click **Finish** to create your project.

Create a Library

To create a library:

1. Select the project name Vaccine Adverse Events to display the project Properties Panel.



Property	Value
Name	Vaccine Adverse Even
Project Start Code	...
Created	4/1/09 3:53 PM
Server	SASApp - Logical Wor
Grid Available	No
Path	C:\EM_Projects\EM6.1
Max. Concurrent Tasks	Default

2. Click the ... button for the Project Start Code property. The Project Start Code dialog box opens.

3. Select the **Code** tab and enter the following code to create a SAS library:

```
libname mylib "c:\vaersdata";
```

Note: The location will depend on where you have stored the data for this tutorial on your system.

4. Click **Run Now**.

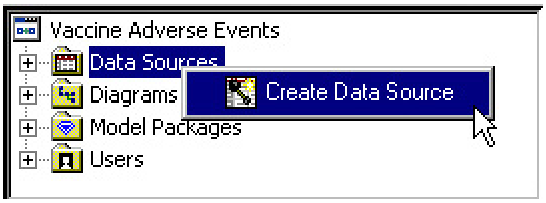
5. Click **OK** to close the Project Start Code dialog box.

Note: An alternate way to create a library is to use the library wizard. To use the library wizard, select **File** ▸ **New** ▸ **Library** from the main menu.

Create a Data Source

To create a data source:

1. Right-click the Data Sources folder in the Project Panel and select **Create Data Source** to open the Data Source wizard.



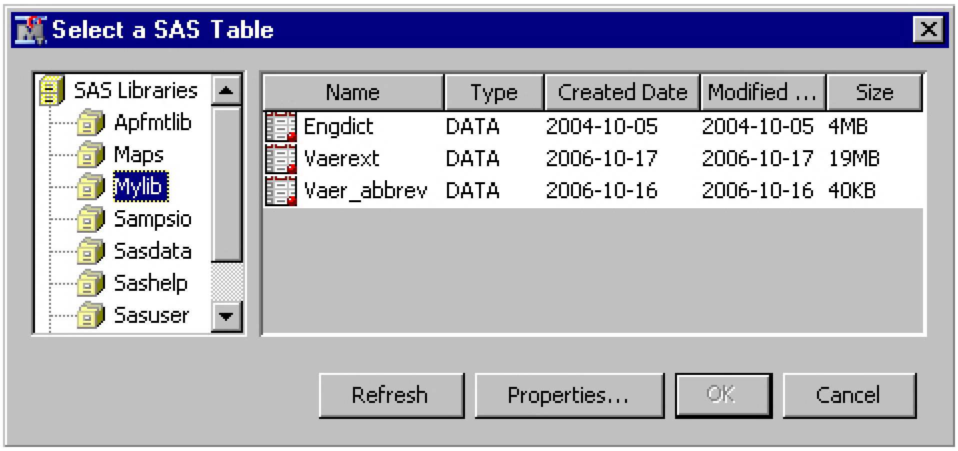
2. Select **SAS Table** in the Source drop-down menu of the Metadata Source page.

3. Click **Next**. The Select a SAS Table page opens.

4. Click **Browse**.

5. Click the SAS library named **Mylib**. The Mylib library folder contents are displayed on the Select a SAS Table dialog box.

   Note: If you do not see SAS data files in the Mylib folder, click **Refresh**.



Select a SAS Table

SAS Libraries
- Apfmtlib
- Maps
- Mylib
- Sampsio
- Sasdata
- Sashelp
- Sasuser

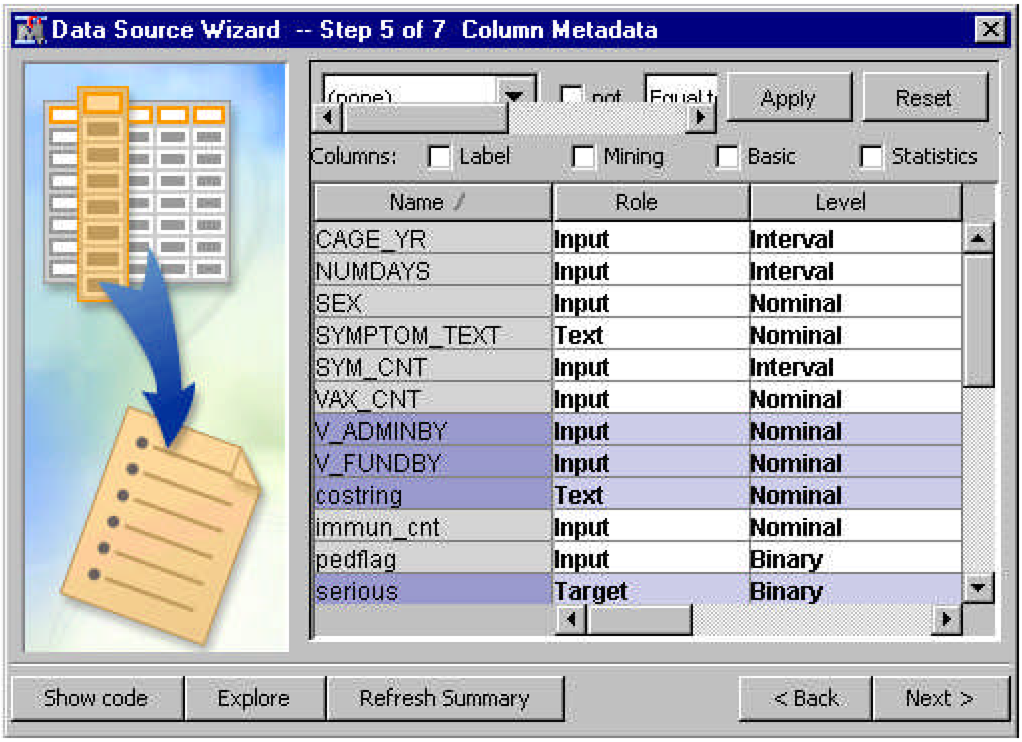
Name	Type	Created Date	Modified ...	Size
Engdict	DATA	2004-10-05	2004-10-05	4MB
Vaerext	DATA	2006-10-17	2006-10-17	19MB
Vaer_abbrev	DATA	2006-10-16	2006-10-16	40KB

6. Select the **VAEREXT** table, and then click **OK**. The two-level name MYLIB.VAEREXT is displayed in the Table box of the Select a SAS Table page.

7. Click **Next**. The Table Information page opens. The Table Properties pane displays metadata for you to review.

8. Click **Next**. The Metadata Advisor Options page opens.

9. Select **Advanced**, and then click **Next**. The Column Metadata page opens.



Data Source Wizard -- Step 5 of 7 Column Metadata

(none) □ not Equal to

Columns: □ Label □ Mining □ Basic □ Statistics

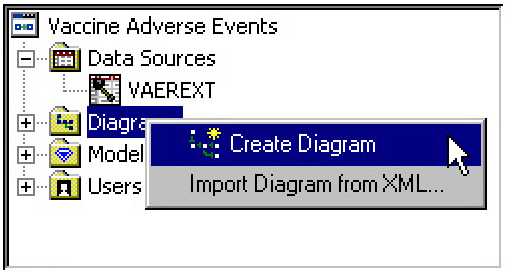
Name	Role	Level
CAGE_YR	Input	Interval
NUMDAYS	Input	Interval
SEX	Input	Nominal
SYMPTOM_TEXT	Text	Nominal
SYM_CNT	Input	Interval
VAX_CNT	Input	Nominal
V_ADMINBY	Input	Nominal
V_FUNDBY	Input	Nominal
costring	Text	Nominal
immun_cnt	Input	Nominal
pedflag	Input	Binary
serious	Target	Binary

10. Select the following variable roles by clicking the role value for each variable value and selecting the indicated value from the drop-down list.
    - Set the role for **V_ADMINBY** to **Input**.
    - Set the role for **V_FUNDBY** to **Input**.
    - Set the role for **costring** to **Text**.
    - Set the role for **serious** to **Target**.
11. Click **Next**. The Decision Configuration page opens.
12. Click **Next**. The Data Source Attributes page opens.
13. Click **Next**. The Summary page opens.
14. Click **Finish**. The VAEREXT table is added to the Data Sources folder in the Project Panel.

Create a Diagram

To create a diagram, complete the following steps:

1. Right-click the Diagram folder in the Project Panel and select **Create Diagram**. The Create New Diagram dialog box opens.



2. Type *VAERS Example* in the Diagram Name box.
3. Click **OK**. The empty VAERS Example diagram opens in the diagram workspace.

Chapter 4
Analyzing the SYMPTOM_TEXT Variable

About the Tasks That You Will Perform

The SYMPTOM_TEXT variable contains the text of an adverse event as it was reported. This chapter explains how you can analyze the SYMPTOM_TEXT variable by performing the following tasks:

1. Identify the VAERS data source with an Input Data node.
2. Partition the input data using the Data Partition node.
3. Set Text Miner node properties using the Properties panel, and run the Text Miner node.
4. View the results using the Interactive Results window.
5. Use the Segment Profile node to examine data segments.

Identify Input Data

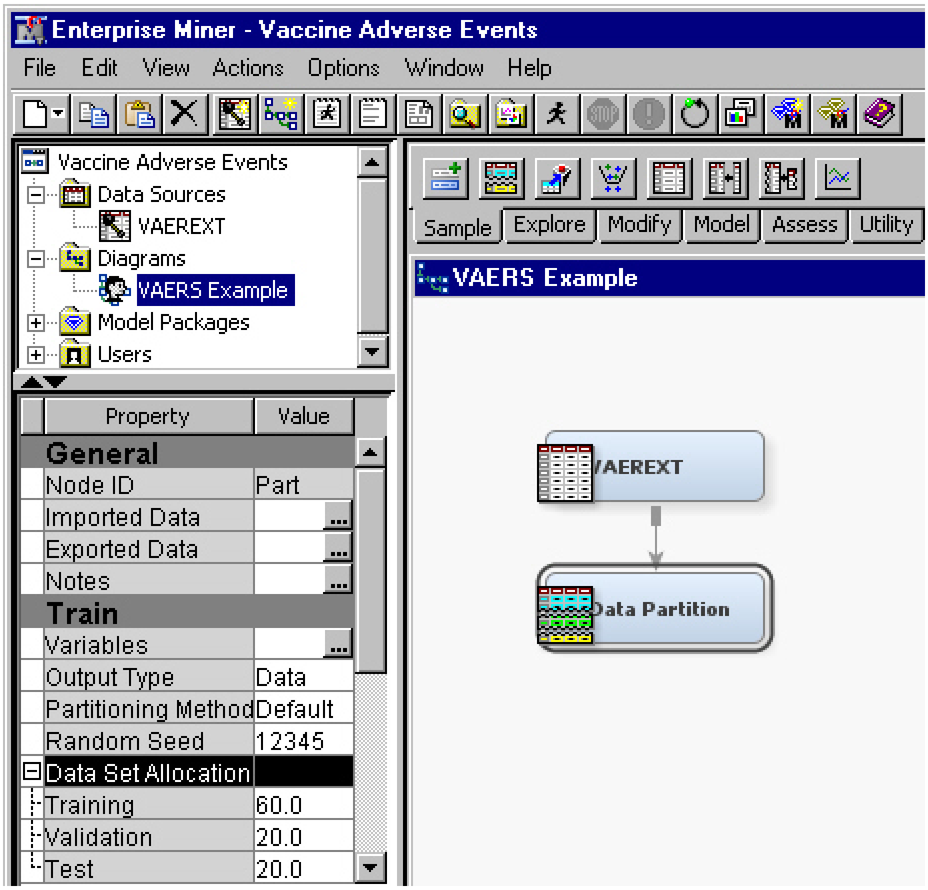
To identify input data:

1. Select the **VAEREXT** data source from the Data Sources list in the Project Panel.
2. Drag and drop VAEREXT into the diagram workspace to create an Input Data node.

Partition Input Data

To partition the input data:

1. Select the **Sample** tab from the node toolbar and drag a Data Partition node into the diagram workspace. Connect the VAEREXT Input Data node to the Data Partition node.



Property	Value
General	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Default
Random Seed	12345
⊟ Data Set Allocation	
Training	60.0
Validation	20.0
Test	20.0

2. Select the Data Partition node to view its properties. Details about the node appear in the Properties Panel. Set the Data Set Allocation properties as follows:

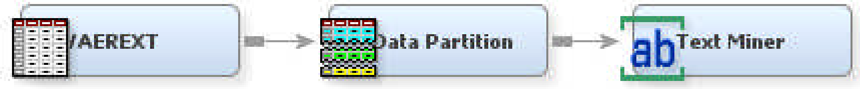
   - Set the **Training** property to **60.0**.
   - Set the **Validation** property to **20.0**.
   - Set the **Test** property to **20.0**.

   These data partition settings will ensure adequate data when you build prediction models with the VAEREXT data.

Set Text Miner Node Properties

To set the Text Miner node properties:

1. Select the **Explore** tab on the toolbar and drag and drop a Text Miner node into the diagram workspace. Connect the Data Partition node to the Text Miner node.



2. Select the Text Miner node to view its properties. Details about the node appear in the Properties Panel. Set the following Parse properties:

   - Set **Terms in Single Document** to **Yes** to include all terms that occur only in a single document.

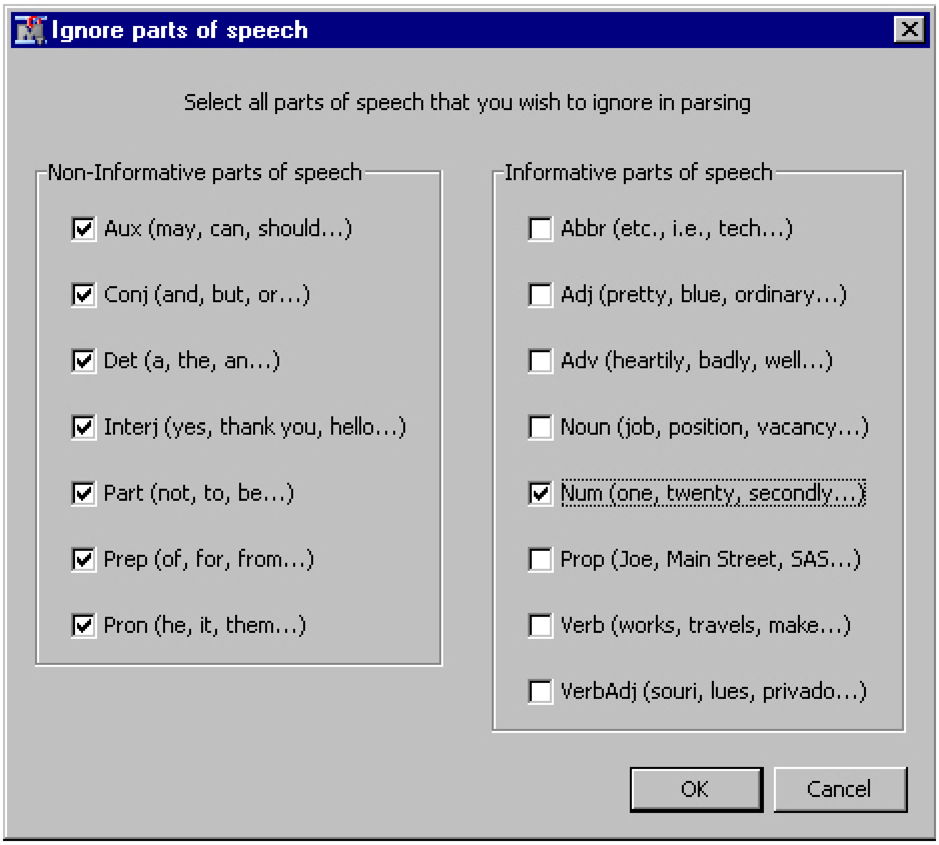
   - Set **Different Parts of Speech** to **No**. For the VAERS data, this setting offers a more compact set of terms.

   - Click the ... button for the Synonyms property. The Select a SAS Table window opens. Select **No data set to be specified**. Click **OK**.

   - Click the ... button for the Ignore Parts of Speech property, and select the following items, which represent parts of speech:

     - **Aux**
     - **Conj**
     - **Det**
     - **Interj**
     - **Part**
     - **Prep**
     - **Pron**
     - **Num**

     Any terms with the parts of speech that you select in the Ignore parts of speech dialog box are ignored during parsing. The selections indicated here ensure that the analysis ignores low-content words such as prepositions and determiners.



Ignore parts of speech

Select all parts of speech that you wish to ignore in parsing

Non-Informative parts of speech

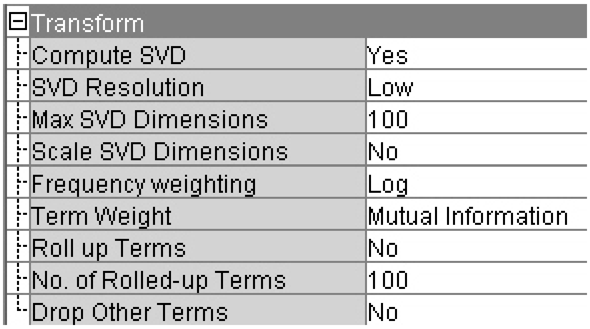
- ☑ Aux (may, can, should...)
- ☑ Conj (and, but, or...)
- ☑ Det (a, the, an...)
- ☑ Interj (yes, thank you, hello...)
- ☑ Part (not, to, be...)
- ☑ Prep (of, for, from...)
- ☑ Pron (he, it, them...)

Informative parts of speech

- □ Abbr (etc., i.e., tech...)
- □ Adj (pretty, blue, ordinary...)
- □ Adv (heartily, badly, well...)
- □ Noun (job, position, vacancy...)
- ☑ Num (one, twenty, secondly...)
- □ Prop (Joe, Main Street, SAS...)
- □ Verb (works, travels, make...)
- □ VerbAdj (souri, lues, privado...)

Click **OK**.

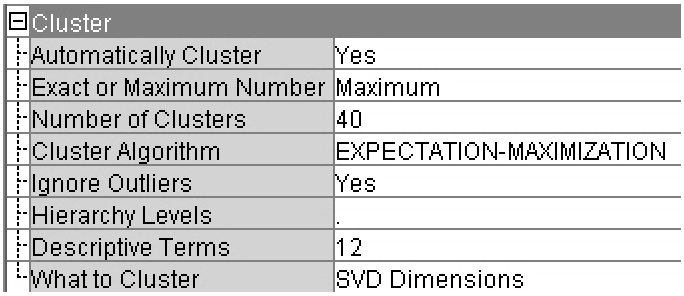
3. Set **Term Weight** to **Mutual Information** so that terms will be differentially weighted when they correspond to serious reactions.



Transform	
Compute SVD	Yes
SVD Resolution	Low
Max SVD Dimensions	100
Scale SVD Dimensions	No
Frequency weighting	Log
Term Weight	Mutual Information
Roll up Terms	No
No. of Rolled-up Terms	100
Drop Other Terms	No

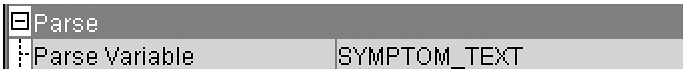
4. Set the following Cluster properties:

   - Set **Automatically Cluster** to **Yes** to answer the question: "What are some categories of reactions that people are experiencing?" You want to categorize these adverse events.
   - Set **Descriptive Terms** to **12** to ease cluster labeling.
   - Set **Ignore Outliers** to **Yes**.



Cluster	
Automatically Cluster	Yes
Exact or Maximum Number	Maximum
Number of Clusters	40
Cluster Algorithm	EXPECTATION-MAXIMIZATION
Ignore Outliers	Yes
Hierarchy Levels	.
Descriptive Terms	12
What to Cluster	SVD Dimensions

5. Right-click the Text Miner node in the diagram workspace, and select **Run**.
6. Click **Yes** in the Confirmation dialog box when you are prompted with a question that asks whether you want to run the path.
7. Click **OK** in the Run Status dialog box that appears after the Text Miner node has finished running. The Text Miner node Parse Variable property has been populated with the SYMPTOM_TEXT variable.

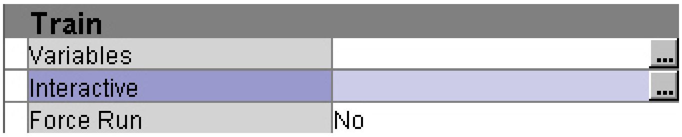


Parse	
Parse Variable	SYMPTOM_TEXT

View Interactive Results

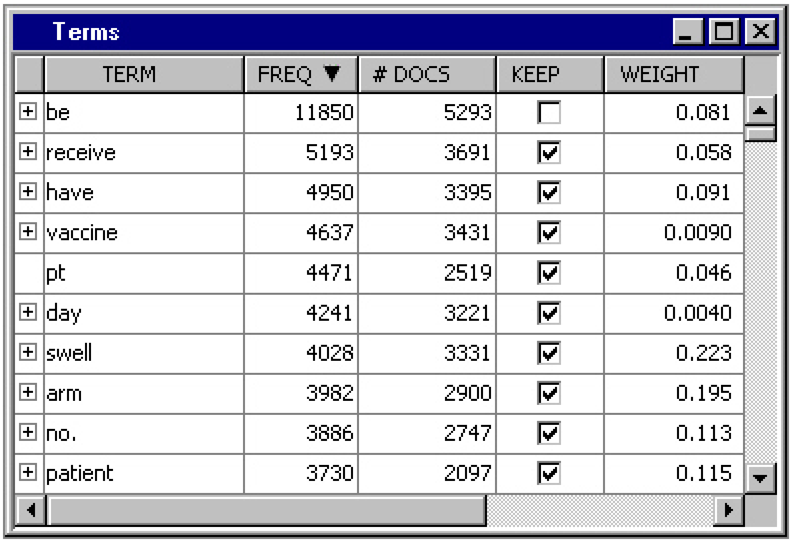
To view interactive results, complete the following steps:

1. Select the Text Miner node, and then Click the ... button for the Interactive property. The Text Miner — Interactive window opens.



Train	
Variables	...
Interactive	...
Force Run	No

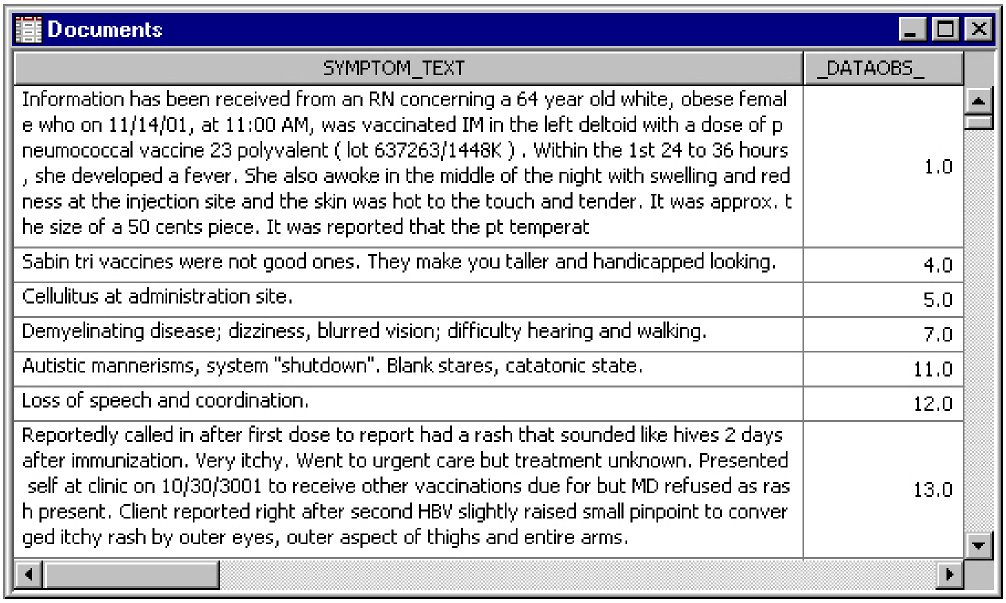
2. View the terms in the Terms window. The terms are sorted in decreasing order of frequency.



Terms

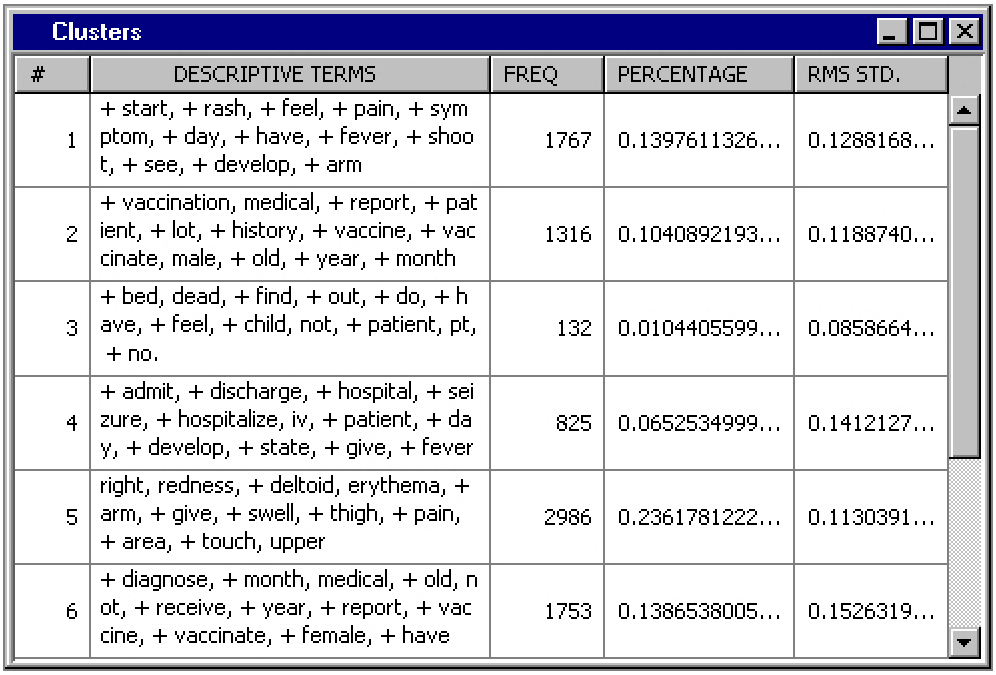
TERM	FREQ ▼	# DOCS	KEEP	WEIGHT
⊞ be	11850	5293	□	0.081
⊞ receive	5193	3691	☑	0.058
⊞ have	4950	3395	☑	0.091
⊞ vaccine	4637	3431	☑	0.0090
pt	4471	2519	☑	0.046
⊞ day	4241	3221	☑	0.0040
⊞ swell	4028	3331	☑	0.223
⊞ arm	3982	2900	☑	0.195
⊞ no.	3886	2747	☑	0.113
⊞ patient	3730	2097	☑	0.115

3. View the documents in the Documents window. Click the Toggle Show Full Text icon on the toolbar to see the full text contained in SYMPTOM_TEXT.



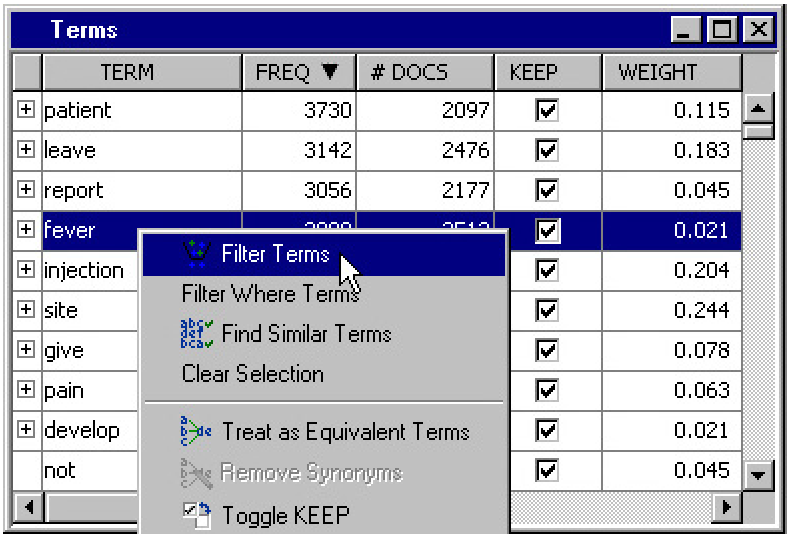
SYMPTOM_TEXT	_DATAOBS_
Information has been received from an RN concerning a 64 year old white, obese female who on 11/14/01, at 11:00 AM, was vaccinated IM in the left deltoid with a dose of pneumococcal vaccine 23 polyvalent (lot 637263/1448K) . Within the 1st 24 to 36 hours , she developed a fever. She also awoke in the middle of the night with swelling and redness at the injection site and the skin was hot to the touch and tender. It was approx. the size of a 50 cents piece. It was reported that the pt temperat	1.0
Sabin tri vaccines were not good ones. They make you taller and handicapped looking.	4.0
Cellulitus at administration site.	5.0
Demyelinating disease; dizziness, blurred vision; difficulty hearing and walking.	7.0
Autistic mannerisms, system "shutdown". Blank stares, catatonic state.	11.0
Loss of speech and coordination.	12.0
Reportedly called in after first dose to report had a rash that sounded like hives 2 days after immunization. Very itchy. Went to urgent care but treatment unknown. Presented self at clinic on 10/30/3001 to receive other vaccinations due for but MD refused as rash present. Client reported right after second HBV slightly raised small pinpoint to converged itchy rash by outer eyes, outer aspect of thighs and entire arms.	13.0

4. View the clusters in the Clusters window.



#	DESCRIPTIVE TERMS	FREQ	PERCENTAGE	RMS STD.
1	+ start, + rash, + feel, + pain, + symptom, + day, + have, + fever, + shoot, + see, + develop, + arm	1767	0.1397611326...	0.1288168...
2	+ vaccination, medical, + report, + patient, + lot, + history, + vaccine, + vaccinate, male, + old, + year, + month	1316	0.1040892193...	0.1188740...
3	+ bed, dead, + find, + out, + do, + have, + feel, + child, not, + patient, pt, + no.	132	0.0104405599...	0.0858664...
4	+ admit, + discharge, + hospital, + seizure, + hospitalize, iv, + patient, + day, + develop, + state, + give, + fever	825	0.0652534999...	0.1412127...
5	right, redness, + deltoid, erythema, + arm, + give, + swell, + thigh, + pain, + area, + touch, upper	2986	0.2361781222...	0.1130391...
6	+ diagnose, + month, medical, + old, not, + receive, + year, + report, + vaccine, + vaccinate, + female, + have	1753	0.1386538005...	0.1526319...

5. Select a term that is related to an adverse reaction that you want to investigate further. For example, select **fever** under the TERM column of the Terms window. Right-click on the term and select **Filter Terms**.



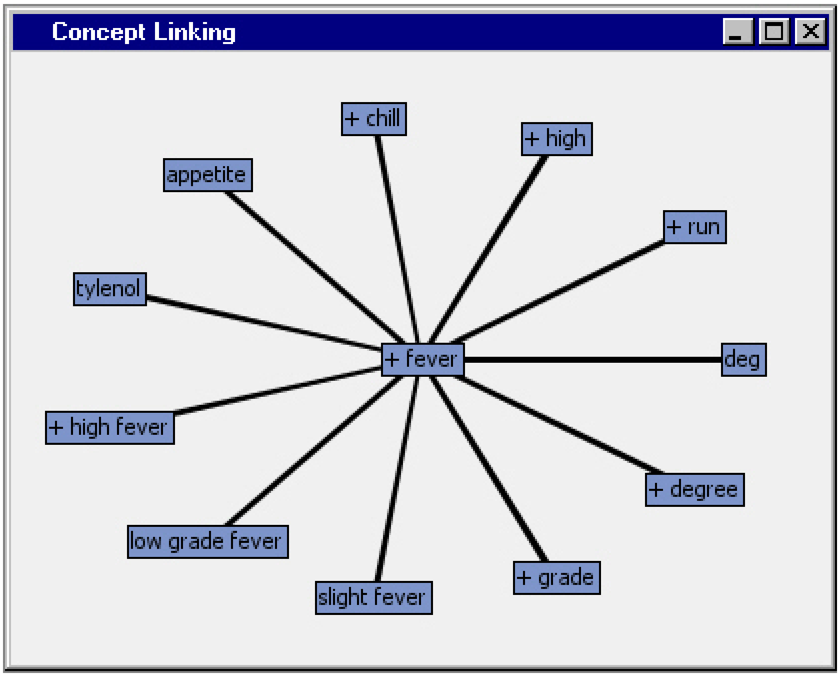
Terms

	TERM	FREQ ▼	# DOCS	KEEP	WEIGHT
⊞	patient	3730	2097	☑	0.115
⊞	leave	3142	2476	☑	0.183
⊞	report	3056	2177	☑	0.045
⊞	fever			☑	0.021
⊞	injection			☑	0.204
⊞	site			☑	0.244
⊞	give			☑	0.078
⊞	pain			☑	0.063
⊞	develop			☑	0.021
	not			☑	0.045

6. Note how the documents displayed and cluster frequencies change. Only those documents containing **fever** are displayed. Moreover, only the documents containing **fever** are counted. If the full text of the document is not shown, click the Toggle Show Full Text icon on the toolbar.

7. Click the Undo icon on the toolbar. This removes the filter that was applied and restores the display that was shown when you opened the Text Miner — Interactive window.

8. Select **fever** in the Terms window, and then right-click **fever** and select **View Concept Links**. The Concept Linking window opens. Concept linking is a way to find and display the terms that are highly associated with the selected term in the Terms table. The selected term is surrounded by the terms that correlate the strongest with it. The Concept Linking window shows a hyperbolic tree graph with fever in the center of the tree structure. It shows you the other terms that are strongly associated with the term fever. To expand the Concept Linking view, right-click on any of the terms that are not in the center of the graph and select **Expand Links**.



9. Look at the clusters in the Clusters window. Can you tell what they are about from the descriptive terms displayed? Do some clusters look vague or unclear?

10. Choose one of the clusters that looks vague or unclear. This is fairly subjective, but, for this example, you can use Cluster 2 as an example of a vague or unclear cluster. Right-click on Cluster 2 and select **Filter Clusters**. This action filters the results to show only those documents and terms that are relevant to Cluster 2. All the documents shown in the Documents window are contained in Cluster 2, and terms are now ordered by frequency within that cluster. Read the text of some of the documents in this cluster. Does this clarify the cluster better?

11. Click the Undo icon ↶ on the toolbar to undo any filters.

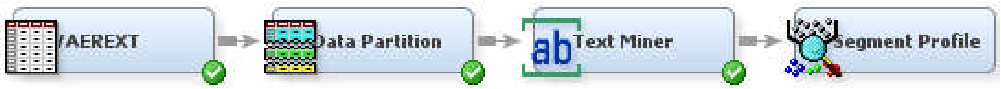
12. Close the Text Miner — Interactive window.

Examine Data Segments

In this section, you will examine segmented or clustered data using the Segment Profile node. A segment is a cluster number derived analytically using SAS Text Miner clustering techniques. The Segment Profile node enables you to get a better idea of what makes each segment unique or at least different from the population. The node generates various reports that aid in exploring and comparing the distribution of these factors within the segments and population.

To examine data segments, complete the following steps:

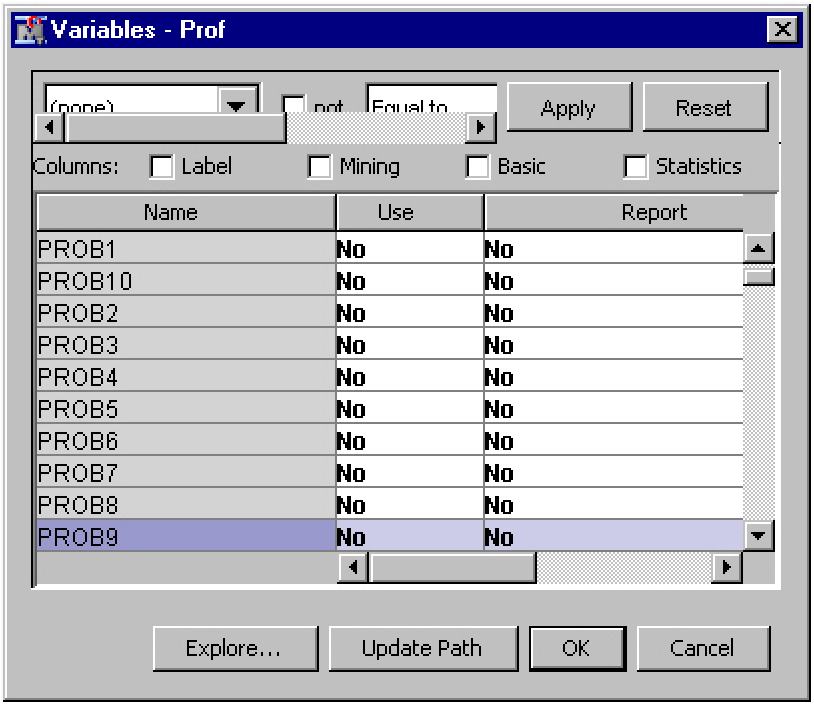
1. From the **Assess** tab, drag and drop a Segment Profile node into the diagram workspace and connect the Text Miner node to the Segment Profile node.



2. Select the Segment Profile node. Select the ... button for the **Variables** property. The Variables — Prof window opens.

3. Select all the PROB variables and set their Use value to **No**.

   Note: You can hold down Shift and select all the PROB variables by clicking on the first PROB variable and dragging the pointer to select all PROB variables. After all PROB variables are selected, you can change the Use value of each selected PROB variable by changing the Use value of one of the PROB variables. This will change the other PROB Use values to the selected value as well.



Name	Use	Report
PROB1	No	No
PROB10	No	No
PROB2	No	No
PROB3	No	No
PROB4	No	No
PROB5	No	No
PROB6	No	No
PROB7	No	No
PROB8	No	No
PROB9	No	No

4. Select all the _SVD_variables and set their **Use** value to **No**.

   Note: You can hold down Shift and select all the _SVD_ variables by clicking on the first _SVD_ variable and dragging the pointer to select all _SVD_ variables. After all _SVD_ variables are selected, you can change the Use value of each selected _SVD_ variable by changing the Use value of one of the _SVD_ variables. This will change the other _SVD_ Use values to the selected value as well.

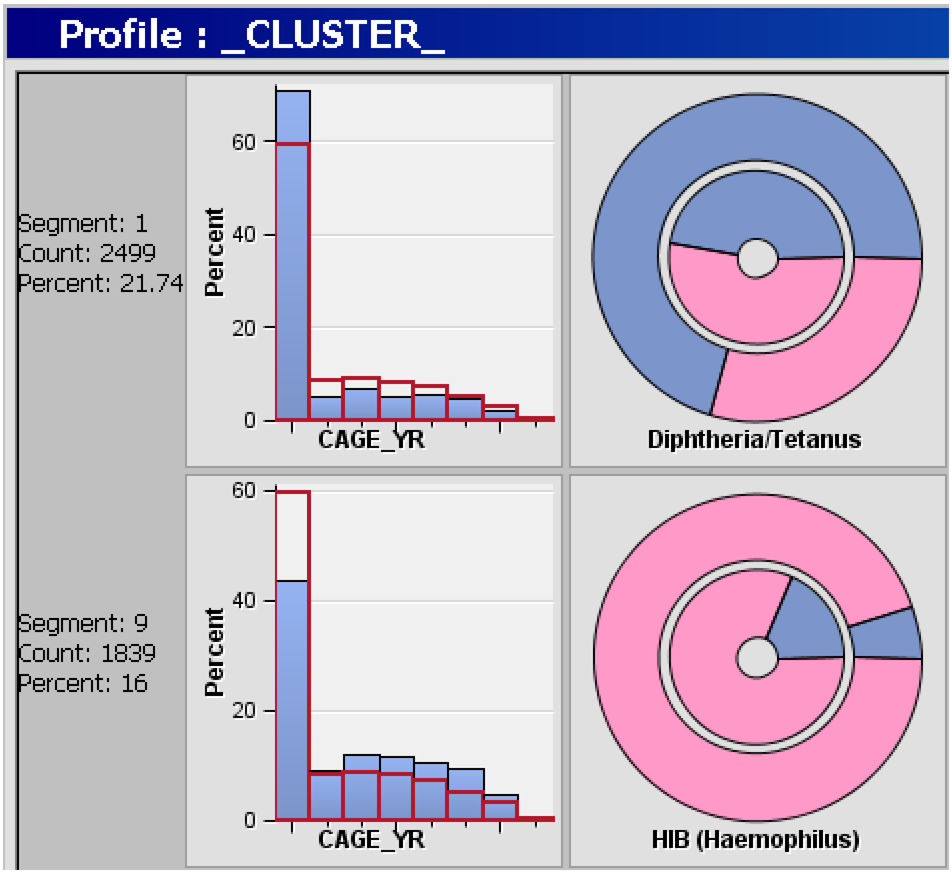
5. Click **OK**.

6. Select the Segment Profile node in the diagram workspace. In the Properties Panel, set the **Minimum Worth** property to **0.0010**.

7. Right-click the Segment Profile node, and select **Run**.

8. Click **OK** in the Confirmation dialog box. After the node finishes running, click **Results** in the Run Status dialog box.

9. Maximize the Profile: _CLUSTER_ window. The following shows a portion of this window.

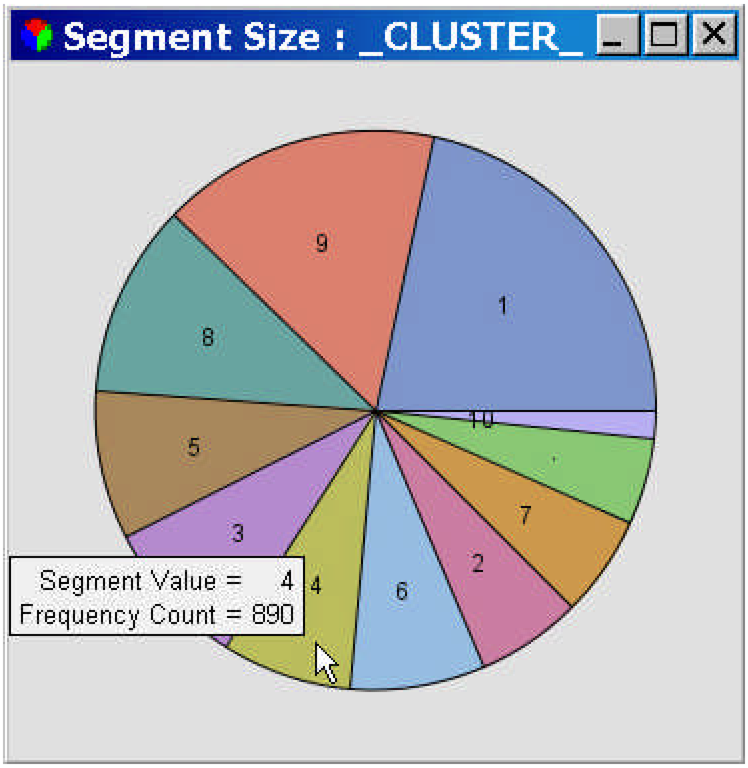


The Profile: _CLUSTER_ window displays a lattice, or grid, of plots that compare the distribution for the identified and report variables for both the segment and the population. The graphs shown in this window illustrate variables that have been identified as factors that distinguish the segment from the population it represents. Each row represents a single segment. The far-left margin identifies the segment, its count, and the percentage of the total population.

The columns are organized from left to right according to their ability to discriminate that segment from the population. Report variables, if specified, appear on the right in alphabetical order after the selected inputs. The lattice graph has the following features:

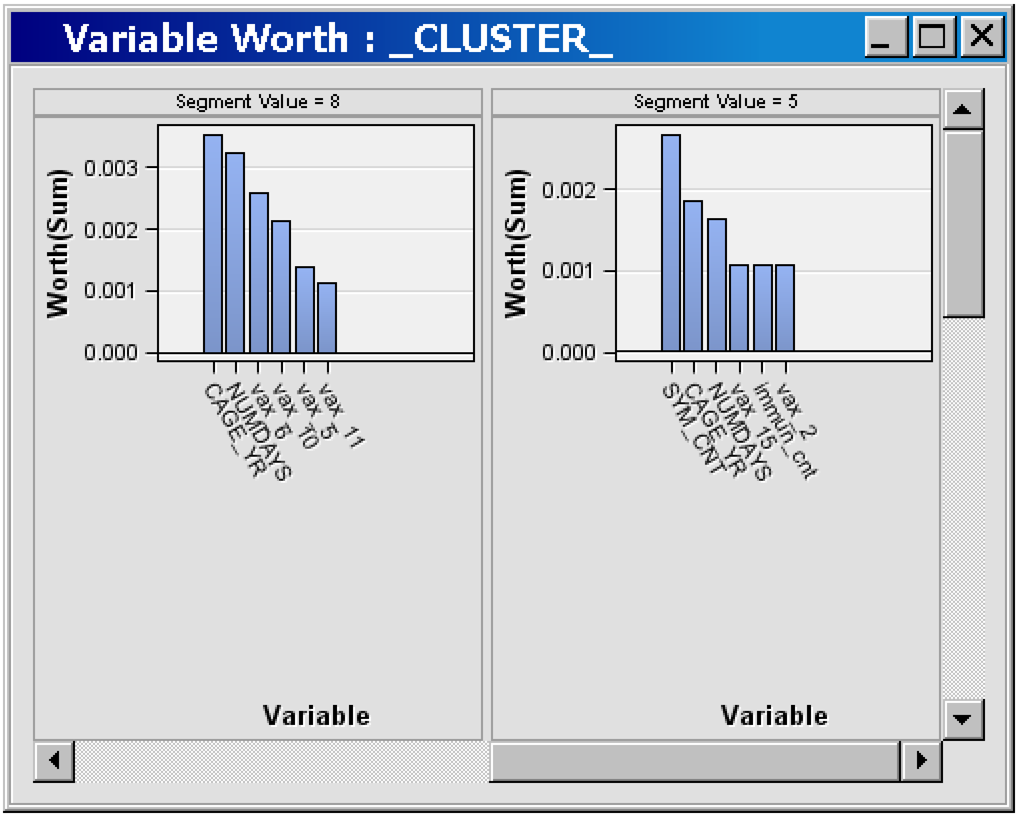
- Class variable — displays as two nested pie charts that consist of two concentric rings. The inner ring represents the distribution of the total population. The outer ring represents the distribution for the given segment.
- Interval variable — displays as a histogram. The blue shaded region represents the within-segment distribution. The red outline represents the population distribution. The height of the histogram bars can be scaled by count or by percentage of the segment population. When you are using the percentage, the view shows the relative difference between the segment and the population. When you are using the count, the view shows the absolute difference between the segment and the population.

10. Maximize the Segment Size: _CLUSTER_ window. The following shows a portion of this window.



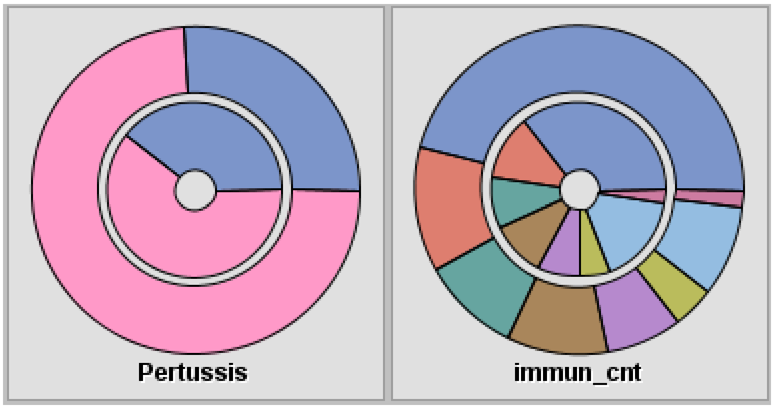
Segment Value = 4
Frequency Count = 890

11. Maximize the Variable Worth: _CLUSTER_ window. The following shows a portion of this window.



12. Note the strong relationships between some of the vaccinations given and the clustered categories. You can think of the "wheels" or concentric rings as follows: the inner circle

represents all the adverse events, while the outer circle contains only the ones in that cluster.



13. Close the **Results** window.

Chapter 5
Cleaning Up Text

About the Tasks That You Will Perform

As demonstrated in the previous chapter, SAS Text Miner does a good job of finding themes that are clear in the data. But, when the data needs cleaning, SAS Text Miner can be less effective at uncovering useful themes. In this chapter, you will encounter manually edited data that contains many misspellings and abbreviations, and you will work on cleaning the data to get better results.

The README.TXT file provided on the VAERS site contains a list of abbreviations commonly used in the adverse event reports. SAS Text Miner enables you to specify a synonym list. A VAER_ABBREV synonym list is provided for you in the Getting Started with SAS Text Miner 4.1 zip file. So that you can create such a synonym list, the abbreviations list from README.TXT was copied into a Microsoft Excel file. The list was manually edited in the Microsoft Excel file and then imported into a SAS data set. For example, CT/CAT was marked as equivalent to computerized axial tomography. For more information about the preprocessing steps, see “Vaccine Adverse Event Reporting System Data Preprocessing” on page 57.

For more information about importing data into a SAS data set, see the following documentation resource:

```
http://support.sas.com/documentation/
```

You will perform the following tasks to clean the text and examine the results:

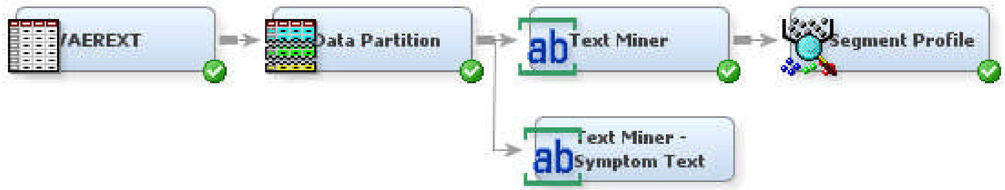
1. Use a synonym data set from the Getting Started with SAS Text Miner 4.1 zip file.
2. Create a new synonym data set using the SAS Code node and the %TEXTSYN macro. The %TEXTSYN macro will run through all the terms, automatically identify which ones are misspellings, and create synonyms that map correctly spelled terms to the misspelled terms.
3. Examine results using merged synonym data sets.

4. Create a stop list to define which words are removed from the analysis. A **stop list** is a collection of low-information or extraneous words that you want to remove from the text, which has been saved as a SAS data set.
5. Explore whether cleaning the text improved the clustering results.

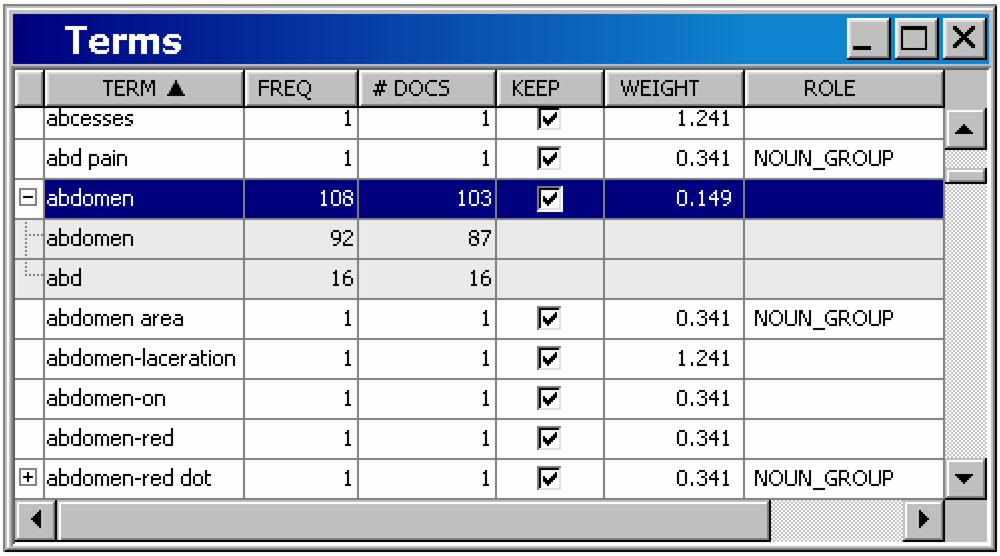
Use a Synonym Data Set

To use a synonym data set:

1. Right-click the Text Miner node in your process flow diagram and select **Copy**. For this example, it is important to copy the node instead of creating a new Text Miner node because the settings you previously specified in the Text Miner node properties panel will be used. Right-click in the empty diagram workspace and select **Paste**.
2. To distinguish this newly pasted Text Miner node from the first node, right-click it, and select **Rename**. Type **Text Miner — Symptom Text** in the Node Name box, and then click **OK**.
3. Connect the Data Partition node to the Text Miner — Symptom Text node.



4. Select the Text Miner — Symptom Text node in the diagram workspace. In the Properties panel, click the ... button for the Synonyms property. The Select a SAS Table dialog box opens.
5. Select the **Mylib** library to view its contents. Select **VAER_ABBREV**, and then click **OK**.
6. Leave all other settings the same as in the original Text Miner node.
7. Right-click the Text Miner — Symptom Text node in the diagram workspace, and select **Run**. Click **OK** in the Confirmation dialog box. Click **OK** in the Run Status dialog box when the node has finished running.
8. Click the ... button for the Interactive property to open the Text Miner — Interactive window.
9. Click the TERM column heading to sort the Terms table.
10. Select **abdomen** under the TERM column in the Terms window. The term abdomen is one of the terms on the right side of the MYLIB.VAER_ABBREV table. In the Terms window, there should be a plus (+) sign next to **abdomen**. Click on the plus sign to expand the term. This shows all synonyms and stems that are mapped to that term. A stem is the root form of a term. Make sure that the child term **abd** is included. Both **abdomen** and **abd** will be treated the same.



Terms

	TERM ▲	FREQ	# DOCS	KEEP	WEIGHT	ROLE
	abcesses	1	1	☑	1.241	
	abd pain	1	1	☑	0.341	NOUN_GROUP
⊟	abdomen	108	103	☑	0.149	
	abdomen	92	87			
	abd	16	16			
	abdomen area	1	1	☑	0.341	NOUN_GROUP
	abdomen-laceration	1	1	☑	1.241	
	abdomen-on	1	1	☑	0.341	
	abdomen-red	1	1	☑	0.341	
⊞	abdomen-red dot	1	1	☑	0.341	NOUN_GROUP

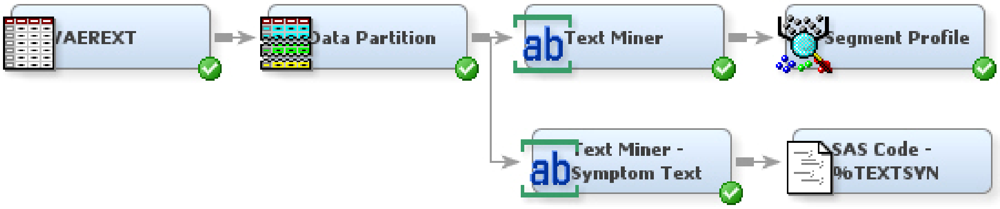
11. Close the Text Miner — Interactive window.

Create a New Synonym Data Set

You can use the SAS Text Miner %TEXTSYN macro to create a new synonym data set. The %TEXTSYN macro evaluates all the terms, automatically identifies which terms are misspellings, and creates synonyms that map correctly spelled terms to misspelled terms.

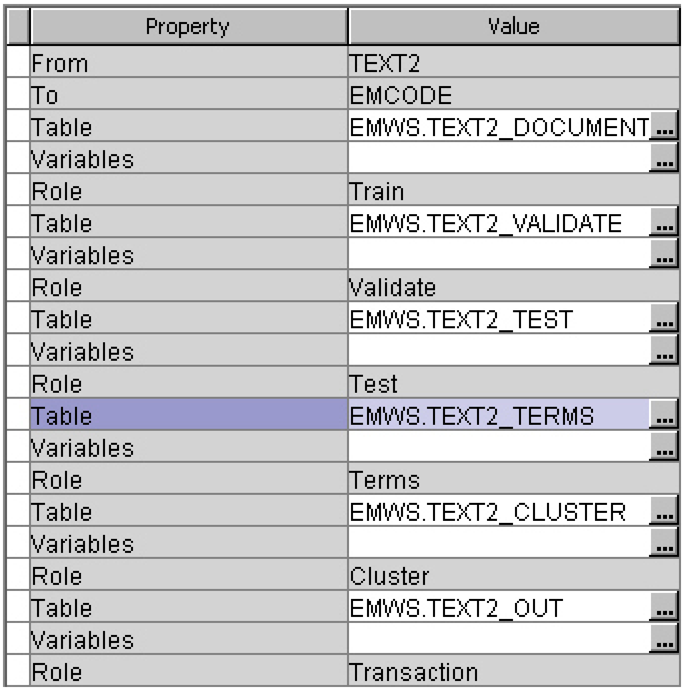
To create a new synonym data set:

1. Select the **Utility** tab and drag a SAS Code node into the diagram workspace. Connect the Text Miner — Symptom Text node to the SAS Code node. Right-click the SAS Code node, and select **Rename**. Type *SAS Code — %TEXTSYN* in the Node Name box. Click **OK**.



2. Select the arrow that connects the Text Miner — Symptom Text node to the SAS Code — %TEXTSYN node. Note the value of the Terms export **Table** property. You will use this value in the TERMDS= parameter in the next step.

   Note: The libref EMWS in the TERMS Table property is dependent upon the diagram number within your SAS Enterprise Miner project. If your diagram is the first one created, then the libref will be EMWS, the second diagram will be EMWS1, the third will be EMWS2, and so on.

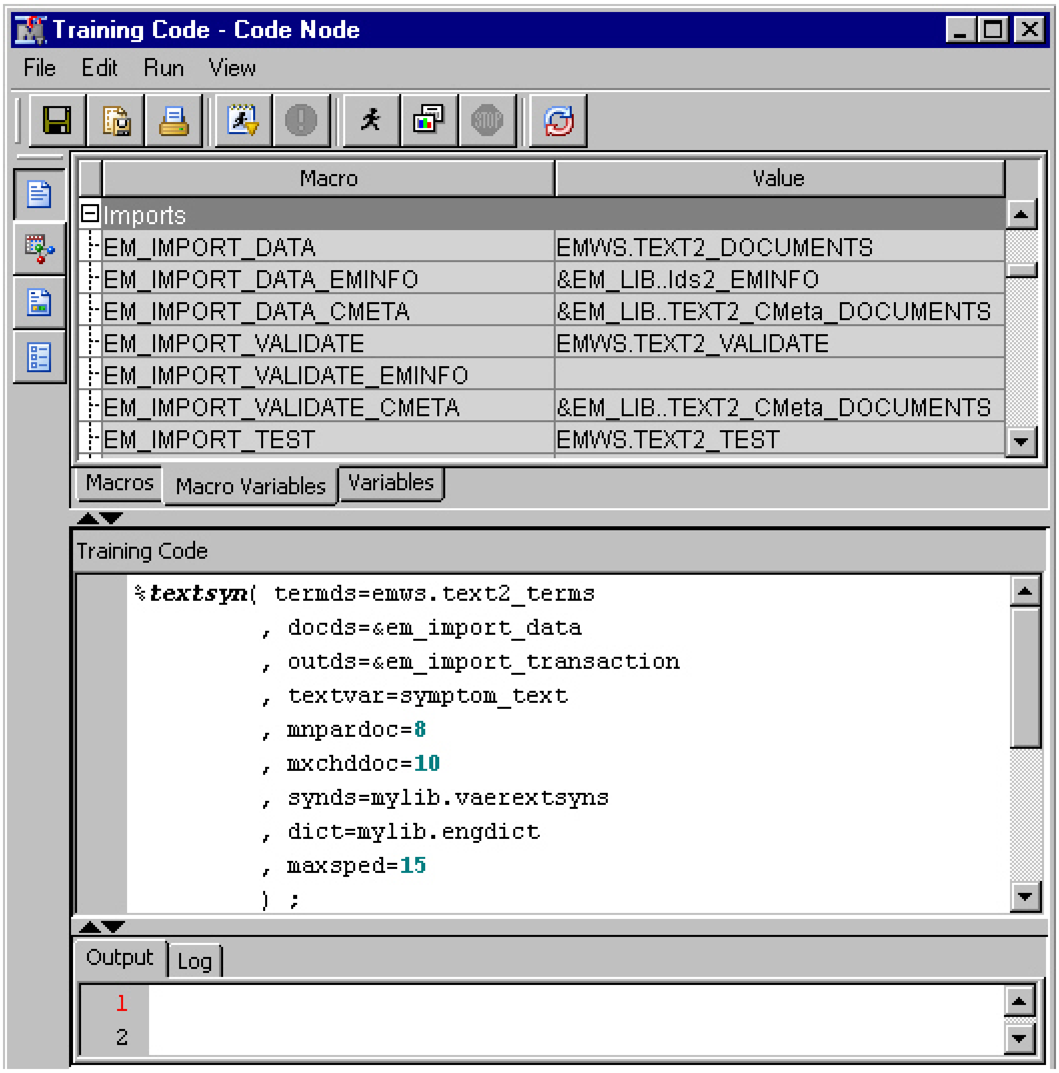


Property	Value
From	TEXT2
To	EMCODE
Table	EMWS.TEXT2_DOCUMENT
Variables	
Role	Train
Table	EMWS.TEXT2_VALIDATE
Variables	
Role	Validate
Table	EMWS.TEXT2_TEST
Variables	
Role	Test
Table	EMWS.TEXT2_TERMS
Variables	
Role	Terms
Table	EMWS.TEXT2_CLUSTER
Variables	
Role	Cluster
Table	EMWS.TEXT2_OUT
Variables	
Role	Transaction

3. Select the SAS Code — %TEXTSYN node, and click the ... button for the **Code Editor** property in the Properties panel.

4. Enter the following code in the Code Editor:

```
%textsyn( termds=emws.text2_terms
         , docds=&em_import_data
         , outds=&em_import_transaction
         , textvar=symptom_text
         , mnpardoc=8
         , mxchddoc=10
         , synds=mylib.vaerextsyns
         , dict=mylib.engdict
         , maxsped=15
         ) ;
```



Macro	Value
Imports	
EM_IMPORT_DATA	EMWS.TEXT2_DOCUMENTS
EM_IMPORT_DATA_EMINFO	&EM_LIB..lds2_EMINFO
EM_IMPORT_DATA_CMETA	&EM_LIB..TEXT2_CMeta_DOCUMENTS
EM_IMPORT_VALIDATE	EMWS.TEXT2_VALIDATE
EM_IMPORT_VALIDATE_EMINFO	
EM_IMPORT_VALIDATE_CMETA	&EM_LIB..TEXT2_CMeta_DOCUMENTS
EM_IMPORT_TEST	EMWS.TEXT2_TEST

```
%textsyn( termds=emws.text2_terms
        , docds=&em_import_data
        , outds=&em_import_transaction
        , textvar=symptom_text
        , mnpardoc=8
        , mxchddoc=10
        , synds=mylib.vaerextsyns
        , dict=mylib.engdict
        , maxsped=15
        ) ;
```

Note: For details on the %TEXTSYN macro, see SAS Text Miner help documentation.

5. Click the button to save the changes.

6. Click the button to run the SAS Code — %TEXTSYN node. Click **Yes** in the Confirmation dialog box.

7. Click **OK** in the dialog box that indicates that the node has finished running.

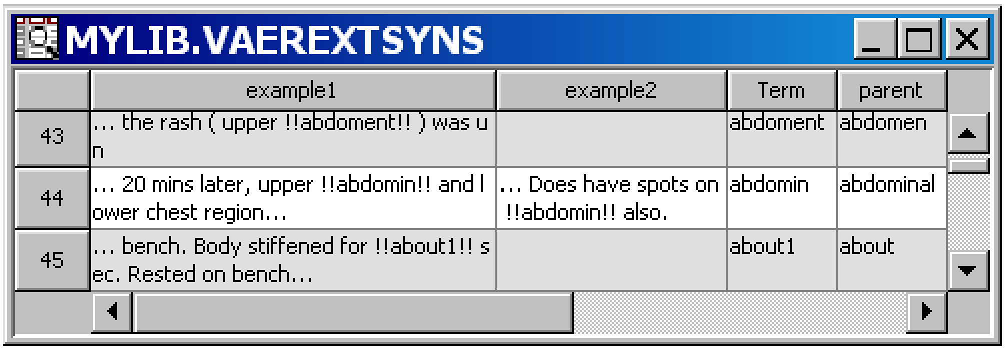
8. Close the Training Code — Code Node window.

9. From the SAS Enterprise Miner window, select **View** ▸ **Explorer**. The Explorer window opens.

10. Click **Mylib**, and then select **Vaerextsyns**.

    Note: If the **Mylib** library is already selected and you do not see the Vaerextsyns data set, you might need to click **Show project data** or refresh the Explorer window to see the **Vaerextsyns** data set.

11. Double-click the Mylib.Vaerextsyns table to examine it.



	example1	example2	Term	parent
43	... the rash (upper !!abdoment!!) was un		abdoment	abdomen
44	... 20 mins later, upper !!abdomin!! and lower chest region...	... Does have spots on !!abdomin!! also.	abdomin	abdominal
45	... bench. Body stiffened for !!about1!! sec. Rested on bench...		about1	about

Here is a list of what the Vaerextsyns columns provide:

- Example1 and example2 are two examples of the term in a document.
- Term is the misspelled word.
- Parent is a guess at the word that was meant.
- Childndocs is the number of documents that contained that term.
- # Documents is the number of documents that contained the parent.
- Minsped is an indication of how close the terms are.
- Dict indicates whether the term is a legitimate English word. Legitimate words can still be deemed misspellings, but only if they occur rarely and are very close in spelling to a frequent target term.

For example, Observation 44 shows **abdomin** to be a misspelling of **abdominal**. Three documents contain **abdomin**, 77 documents contain the parent, **abdomin** is not a legitimate English word, and an example text that contains that misspelling is **20 mins later, upper !!abdomin!!**. Note that double exclamation marks (!!) both precede and succeed the child term in the example text so you can see the term in context.

12. Examine the Vaerextsyns table to see whether you disagree with some of the choices made. For this example, however, assume that the %TEXTSYN macro has done a good enough job detecting misspellings.

    Note: The Vaerextsyns table can be edited using any SAS table editor. You cannot edit this table in the SAS Enterprise Miner GUI. You can change a parent for any misspellings that appear incorrect or delete a row if the Term column contains a valid term.

13. Close the Mylib.Vaerextsyns table and the Explorer window.

Examine Results Using Merged Synonym Data Sets

In this set of tasks, you can create a new data set that contains all the observations from both the Mylib.Vaerextsyns and Mylib.Vaer_abbrev data sets, and examine the results using the merged synonym data set. Complete the following steps:

1. Select the **Utility** tab and drag a SAS Code node into the diagram workspace. Connect the SAS Code — %TEXTSYN node to the new SAS Code node. Right-click the new SAS Code node, select **Rename**, and type *SAS Code — Merge SL*, where *SL* stands for *Synonym Lists*, in the Node Name box. Click **OK**.

2. Select the SAS Code — Merge SL node and click the ... button for the **Code Editor** property. The Code Editor opens.

3. Enter the following code in the Code Editor:

```
data mylib.vaerextsyns_new;
     set mylib.vaerextsyns mylib.vaer_abbrev;
run;
```

   This code merges the resulting synonyms data set from the first SAS Code — %TEXTSYN node with the abbreviations data set.

4. Click the 💾 button to save changes. Close the Code Editor window.

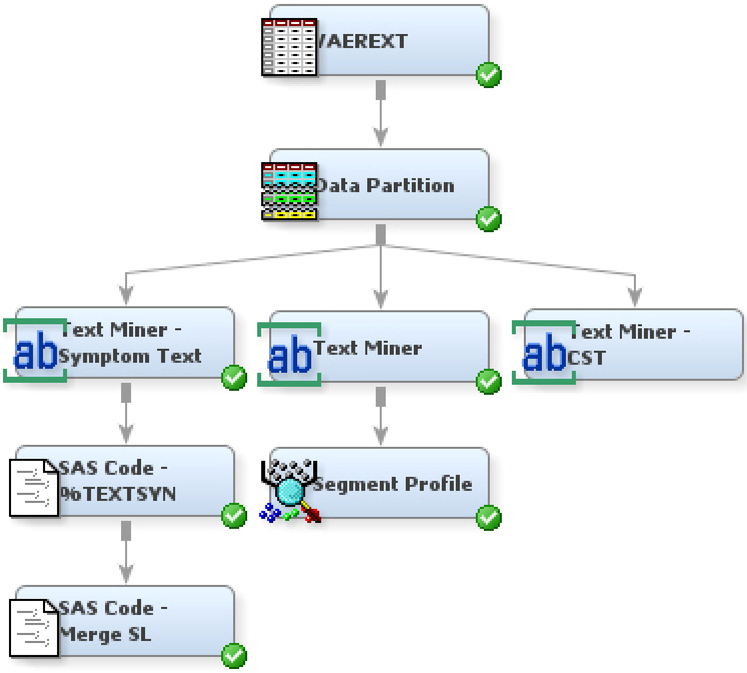
5. Right-click the SAS Code — Merge SL node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **Results** in the Run Status dialog box when the node has finished running.

6. From the Results window, select **View** ▸ **SAS Results** ▸ **Log** to see the SAS code where the new data set is created.

   Close the Results window.

7. Right-click the Text Miner — Symptom Text node and select **Copy** from the menu. Right-click an empty space in the diagram workspace and select **Paste**. It is important to copy the Text Miner — Symptom Text node instead of creating a new Text Miner node in order to keep the same property settings you previously configured for the Text Miner — Symptom Text node. Right-click the new Text Miner node, and select **Rename**. Type *Text Miner — CST*, where *CST* stands for *Cleaned Symptom Text*, in the Node Name box. Click **OK**.

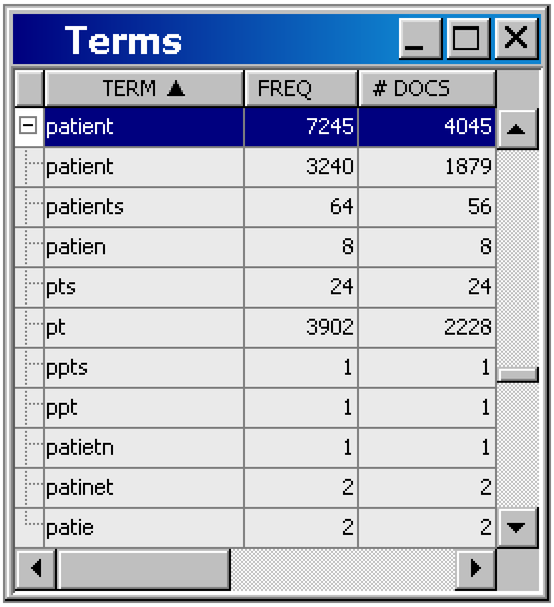
8. Connect the Data Partition node to the Text Miner — CST node.



9. Select the Text Miner — CST node. Set the following properties in the Properties Panel for the Text Miner — CST node:

- Click the ... button for the **Synonyms** property. Select **Mylib** to display its contents if it is not already selected. Click **Refresh**. Select **Mylib.Vaerextsyn_new** from the Select a SAS Table window. Click **OK**.
- Set **Terms in a Single Document** to **No**.

10. Right-click the Text Miner — CST node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **OK** in the Run Status dialog box when the node has finished running.
11. Click the ... button for the **Interactive** property in the Text Miner — CST node Properties Panel. The Interactive Results window opens.
12. Select the plus sign (+) next to **patient** in the Terms table. Note that the misspellings **patien**, **patietn**, and **patie** are included as child terms.



Terms

	TERM ▲	FREQ	# DOCS
⊟	patient	7245	4045
	patient	3240	1879
	patients	64	56
	patien	8	8
	pts	24	24
	pt	3902	2228
	ppts	1	1
	ppt	1	1
	patietn	1	1
	patinet	2	2
	patie	2	2

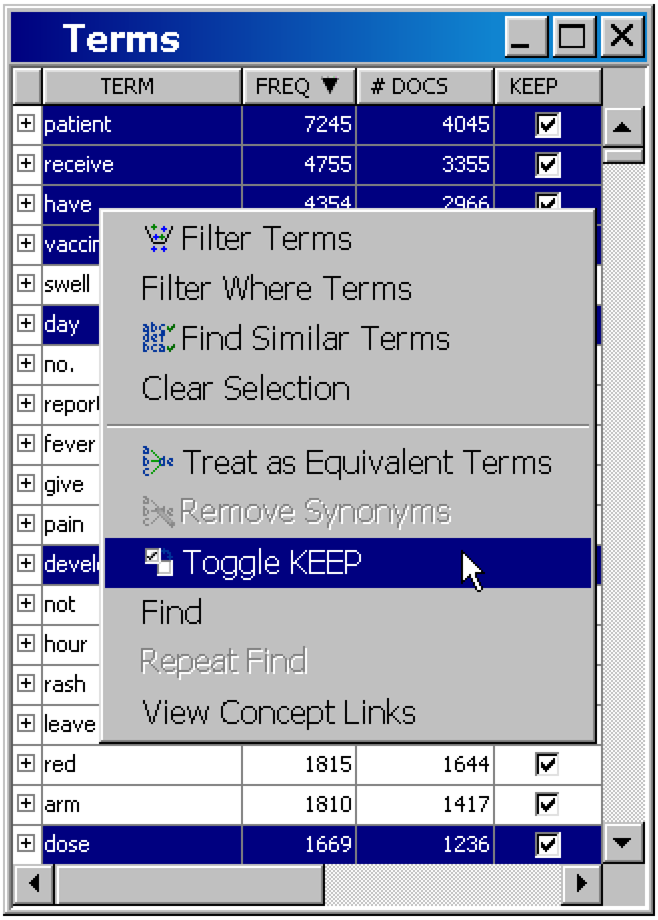
Create a Stop List

A stop list is a simple collection of low-information or extraneous words that you want to remove from the text, which has been saved as a SAS data set.

To create a stop list:

1. Click the FREQ column heading to sort the Terms table by frequency. Make sure that the Freq label has an arrow that points downward to indicate that the Freq column is sorted in descending order.
2. Drop some terms that have no bearing on what the adverse reaction is. Hold down the CTRL key and click on these terms: **patient**, **have**, **receive**, **vaccine**, **day**, **develop**, and **dose**. Right-click to open the menu. Select **Toggle KEEP** to uncheck the **Keep** attribute. This removes the checkmark from the Keep column for each term you have selected.

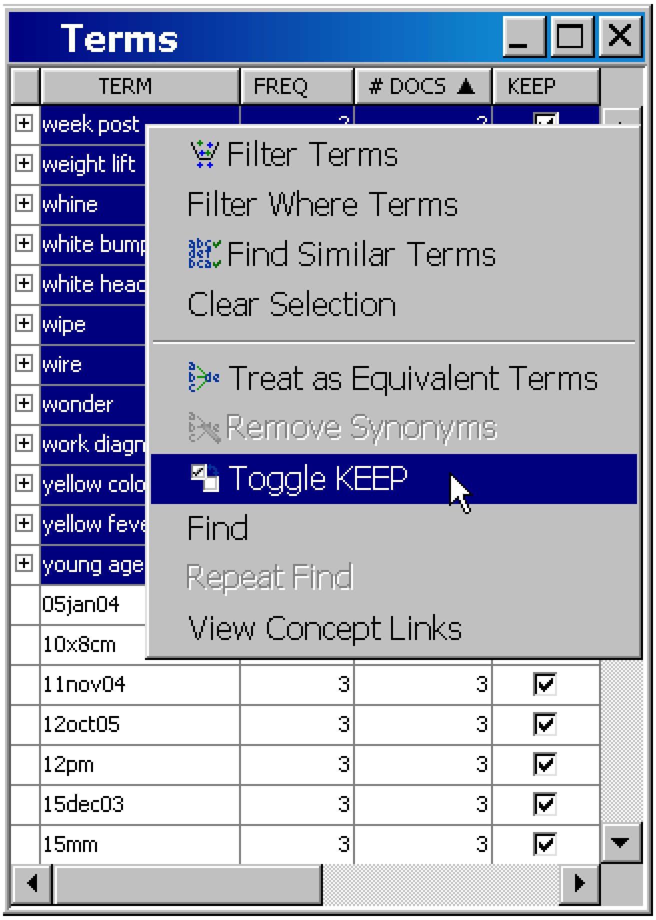
   There are several more terms you could choose to exclude. Only a few are itemized here to demonstrate the concept and process. If additional terms are dropped from the analysis, note that different results will be obtained that will not match those later in this document.



TERM	FREQ ▼	# DOCS	KEEP
⊞ patient	7245	4045	☑
⊞ receive	4755	3355	☑
⊞ have	4354	2966	☑
⊞ vaccir			
⊞ swell			
⊞ day			
⊞ no.			
⊞ reporl			
⊞ fever			
⊞ give			
⊞ pain			
⊞ devel			
⊞ not			
⊞ hour			
⊞ rash			
⊞ leave			
⊞ red	1815	1644	☑
⊞ arm	1810	1417	☑
⊞ dose	1669	1236	☑

3. Click the # DOCS column heading and ensure that the sort arrow is pointing upward. This sorts the terms by count.

4. Click and drag the mouse to select all terms with counts of 2. Right-click a selected term and select **Toggle KEEP** to drop these terms from the analysis.



TERM	FREQ	# DOCS ▲	KEEP
week post			
weight lift			
whine			
white bump			
white head			
wipe			
wire			
wonder			
work diagn			
yellow colo			
yellow feve			
young age			
05jan04			
10x8cm			
11nov04	3	3	☑
12oct05	3	3	☑
12pm	3	3	☑
15dec03	3	3	☑
15mm	3	3	☑

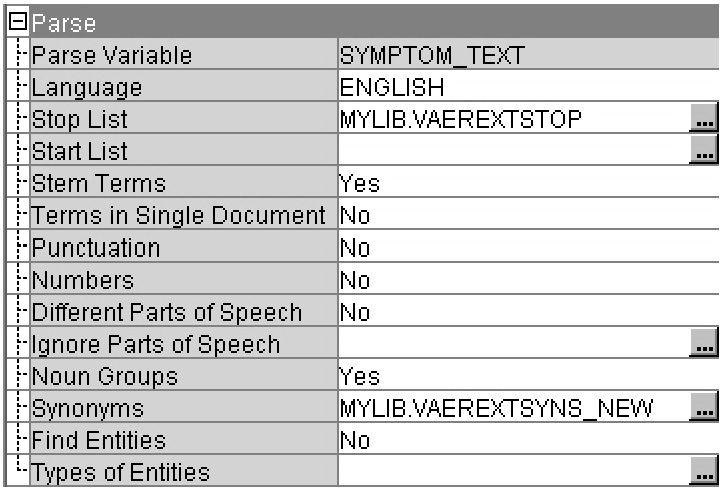
5. Select **File** ▸ **Save Stop List**.

6. Select the **Mylib** library and type *VAEREXTSTOP* in the **Data Set Name** box.

7. Click **OK**.

8. Close the Text Miner — Interactive window.

9. Note that the **Stop List** property of the Text Miner — CST node is set to **MYLIB.VAEREXTSTOP**.

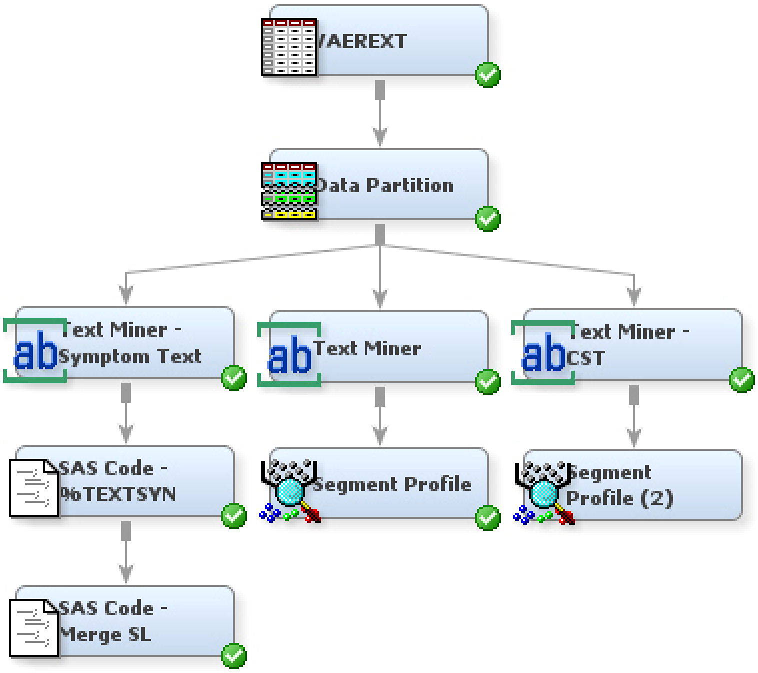


Parse	
Parse Variable	SYMPTOM_TEXT
Language	ENGLISH
Stop List	MYLIB.VAEREXTSTOP ...
Start List	...
Stem Terms	Yes
Terms in Single Document	No
Punctuation	No
Numbers	No
Different Parts of Speech	No
Ignore Parts of Speech	...
Noun Groups	Yes
Synonyms	MYLIB.VAEREXTSYNS_NEW ...
Find Entities	No
Types of Entities	...

Explore Results Improvements

You can redo clustering to explore the improvements to results from cleaning the SYMPTOM_TEXT variable. Complete the following steps:

1. Verify that the Cluster property settings for the Text Miner — CST node are the same as in previous examples.
2. Right-click the Text Miner — CST node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **OK** in the Run Status dialog box when the node has finished running.
3. Select the ... button for the Interactive property in the Text Miner — CST node property panel to open the Interactive Results window. Look at the Clusters table.
4. Compare these results to the first Text Miner node results from "View Interactive Results" on page 21. Does the clustering seem to have improved with the cleaned SYMPTOM_TEXT data?
5. Close the Interactive Results window.
6. Right-click the Segment Profile node, and select **Copy** from the menu. Right-click an empty space in the diagram workspace, and select **Paste** from the menu.
7. Connect the Text Miner — CST node to the Segment Profile (2) node.



8. Click the ... for the **Variables** property for the Segment Profile (2) node to open the Variables — Prof2 window. Make sure that the PROB variables and the _SVD_ variables have a Use value of **No**.
9. Click **OK** to save the variables settings and close the Variables — Prof2 window.
10. Right-click the Segment Profile (2) node and select **Run**. Click **Yes** in the Confirmation dialog box.

11. Click **Results** in the Run Status dialog box to open the Results window when the node has finished running. Note the significant relationships in the table. Do the relationships appear clearer with the cleaned text than they did with the uncleaned text?

12. Close the Results window.

Chapter 6
Predictive Modeling with Text Variables

About the Tasks That You Will Perform

Long before text mining, researchers have needed to analyze text. In the field of drug trials, the need was acute enough that coding systems were developed to automatically pull out keywords or synonyms of keywords that could then be analyzed to understand adverse events. The COSTART coding system was one such attempt. COSTART terms consist of one to three tokens: a symptom, an optional body part, and an optional subpart. One initial task is to find what factors influence whether a reaction becomes serious and how well these factors are captured by the COSTART terms. One way of doing this is to use SAS Text Miner to see how well the COSTART terms predict the seriousness of the adverse event. This chapter explores an example of predictive modeling in SAS Text Miner.

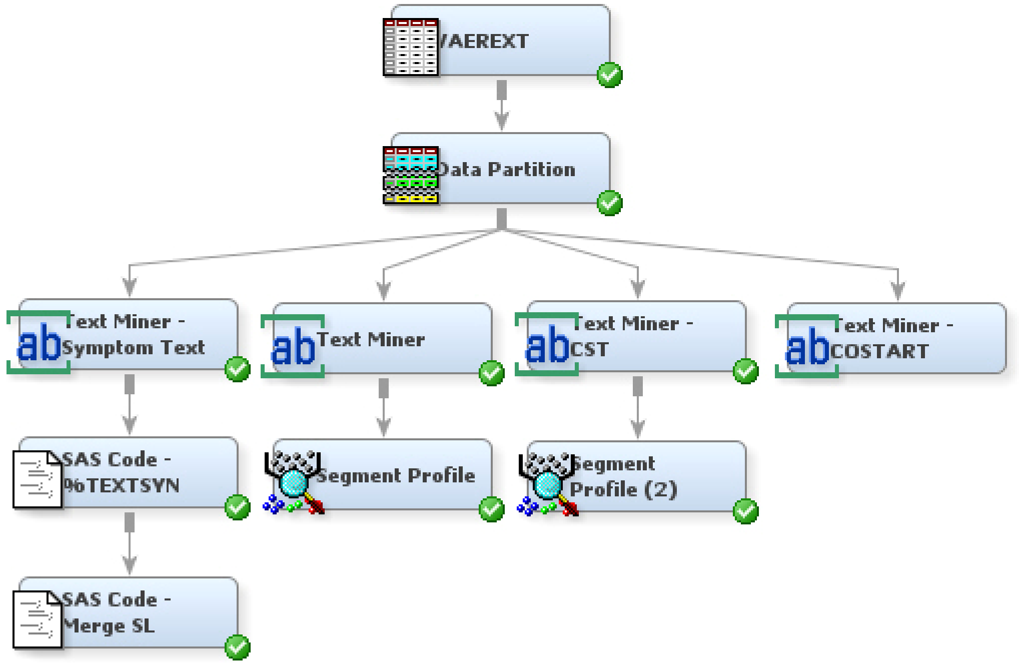
To analyze texts with predictive models, you will perform the following tasks:

1. Use the COSTRING variable and the Decision Tree node to create a model.
2. Use the SYMPTOM_TEXT variable and the Decision Tree node to create a model.
3. Compare the models using the Model Comparison node.

Use the COSTRING Variable to Model

To use the COSTRING variable to create a model:

1. Select the **Explore** tab on the toolbar and drag and drop a Text Miner node into the diagram workspace. Connect the Data Partition node to the Text Miner node.
2. Right-click the new Text Miner node and select **Rename**. Type *Text Miner — COSTART* in the Node Name box, and click **OK**.



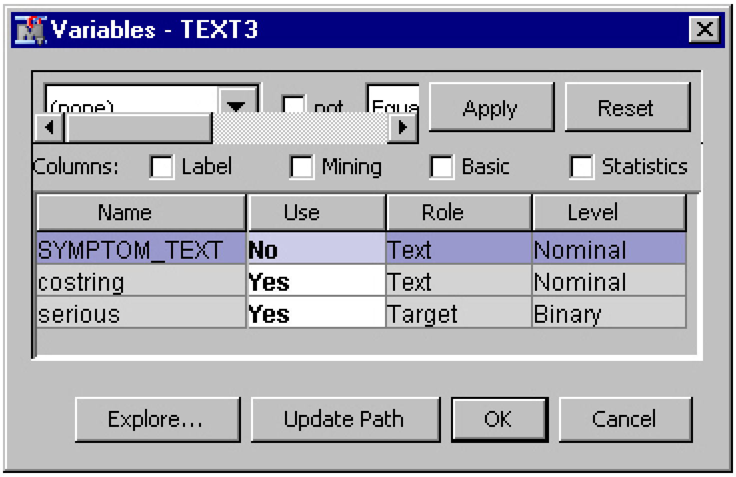
3. Select the VAEREXT node in the diagram workspace. Click the ... button for the **Variables** property in the Properties Panel for the VAEREXT node.

   Recall that there were two text variables, COSTRING and SYMPTOM_TEXT, from the initial data source. By default, SAS Text Miner will use the longer text variable, SYMPTOM_TEXT. In this chapter, you want to mine the COSTRING variable.

   Click **OK** to close the Variables window.

4. Select the Text Miner — COSTART node. Set the following properties in the Properties Panel for the Text Miner — COSTART node:

   - Click the ... button for the **Variables** property. In the Variables window, set the **Use** value for the **SYMPTOM_TEXT** variable to **No**, the **Use** value for the **costring** variable to **Yes**, and the **Use** value for **serious** variable to **Yes**. Click **OK** to save your changes.



Name	Use	Role	Level
SYMPTOM_TEXT	No	Text	Nominal
costring	Yes	Text	Nominal
serious	Yes	Target	Binary

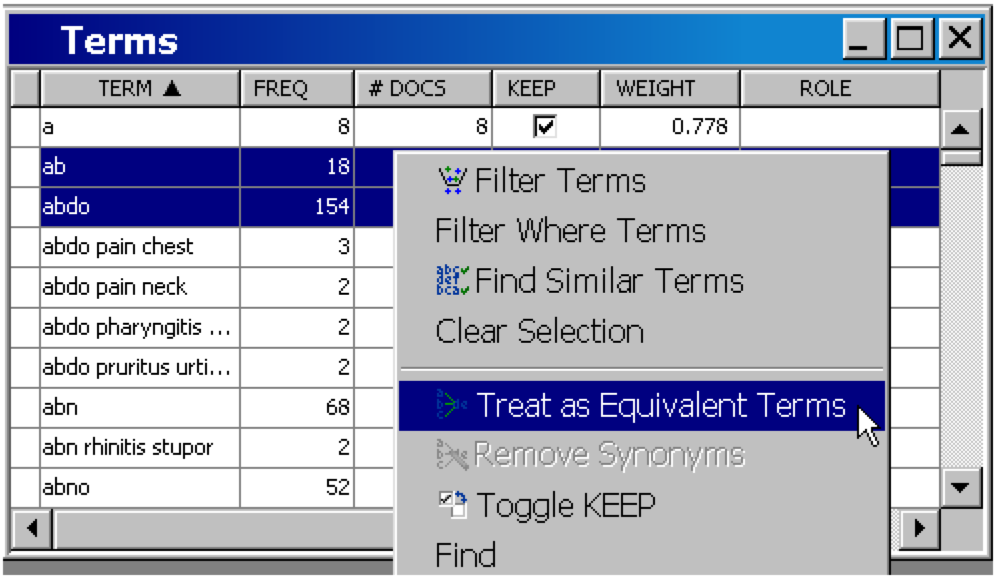
- Click the ... button to the right of the **Stop List** property. Select the **No data set to be specified** check box in the Select a SAS Table dialog box. This removes the entry for the stop list so that no stop list is used. Click **OK**.
- Set **Different Parts of Speech** to **No**.

5. Right-click the Text Miner — COSTART node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **OK** in the Run Status dialog box when the node has finished running.

6. In the Properties Panel, make sure that the **Parse Variable** property of the Text Miner — COSTART Terms node is set to **costring**.

7. Click the ... button for the **Interactive** property to open the Interactive Results window. One problem with COSTART is that it does not always use the same keyword to describe the same term or equivalent terms. For example, **abdomen** is shown in COSTART as **ab** and as **abdo**. Sometimes there are modifiers that you do not need. You could run the %TEXTSYN macro, but because these are abbreviations, the macro will probably not find all of the correct spellings. You need to manually clean some terms.

8. Sort the terms in the Terms window by clicking on the Term column heading. Select **ab** and **abdo** from the TERM column. Right-click and select **Treat as Equivalent Terms**.



TERM ▲	FREQ	# DOCS	KEEP	WEIGHT	ROLE
a	8	8	☑	0.778	
ab	18				
abdo	154				
abdo pain chest	3				
abdo pain neck	2				
abdo pharyngitis ...	2				
abdo pruritus urti...	2				
abn	68				
abn rhinitis stupor	2				
abno	52				

Select **abdo** from the Create Equivalent Terms dialog box. Click **OK**.

Look through the data set and create synonyms by holding the CTRL or Shift keys and clicking the terms that you consider to be the same. Then, right-click on these selected terms and select **Treat as Equivalent Terms**.

9. Repeat this process as many times as you need. It might be helpful to filter the terms so that you can view the full text of COSTART before combining terms.

10. Select **File** ▸ **Save Synonyms** from the Interactive Results window menu. Select **Mylib** in the drop-down menu for the library field, and type *COSTARTSYNS* in the Data Set Name field. Click **OK**.

11. Close the Text Miner — Interactive window.

12. Note that the **Synonyms** property in the Properties Panel has been set to the new MYLIB.COSTARTSYNS synonym data set.

13. COSTART terms should represent keywords, so you want to create variables for each keyword. Set the following **Transform** properties in the Properties Panel:

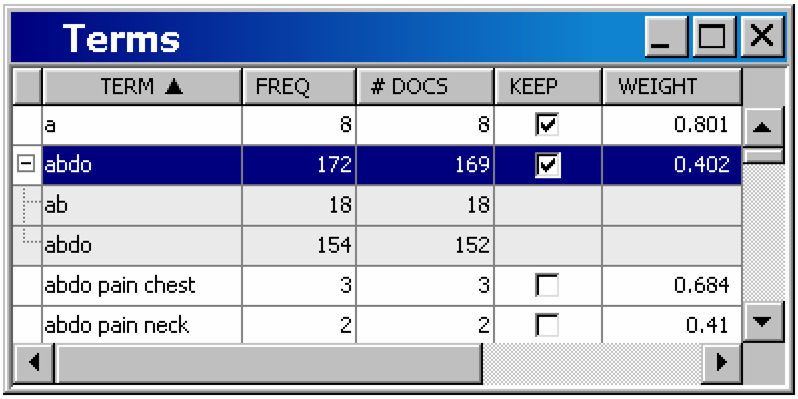
    - Set **Compute SVD** to **No**.
    - Set **Term Weight** to **Mutual Information**.
    - Set **Roll up Terms** to **Yes**.
    - Set **No. of Rolled-up terms** to **400**.
    - Set **Drop Other Terms** to **Yes**.

14. Right-click the Text Miner — COSTART node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **OK** in the Run Status dialog box when the node has finished running.

15. Click the ... button for the Interactive property to open the Text Miner — Interactive window and view the Terms window.

16. Sort the TERM column until the arrow on the column heading is pointing up.

    Note: Terms with a plus (+) sign indicate the synonyms you have specified. Click the plus (+) sign to expand the child terms underneath the respective parent term.

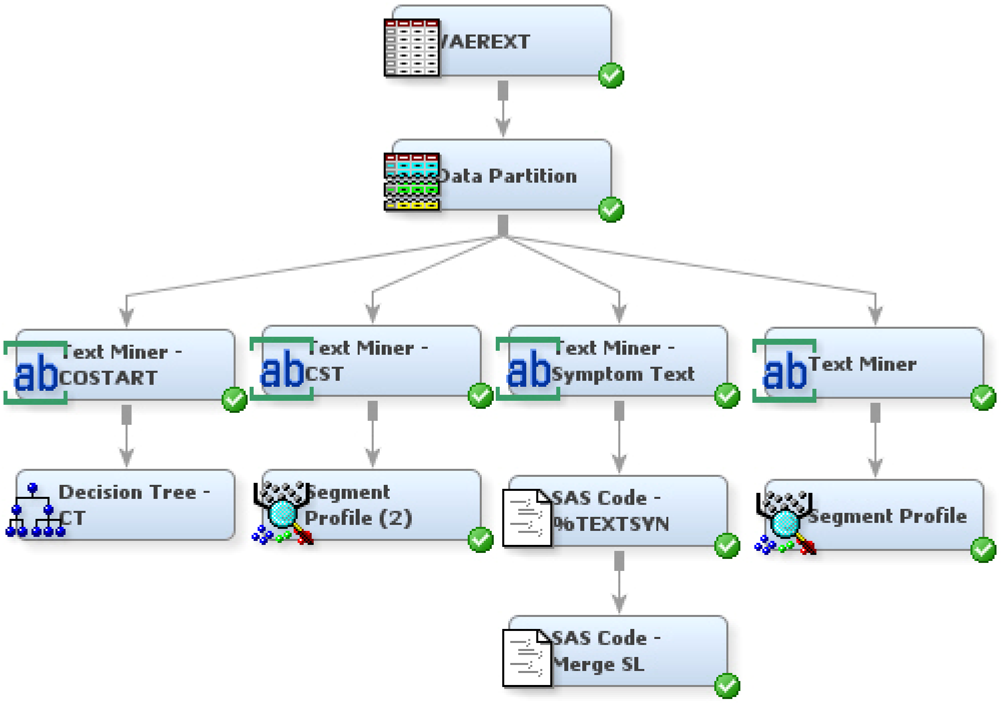


	TERM ▲	FREQ	# DOCS	KEEP	WEIGHT
	a	8	8	☑	0.801
⊟	abdo	172	169	☑	0.402
	ab	18	18		
	abdo	154	152		
	abdo pain chest	3	3	□	0.684
	abdo pain neck	2	2	□	0.41

17. Scroll down until you see terms that do not have a checkmark beneath the Keep column. A separate variable will not be created for these terms. They were not considered significant enough (based on rolling up only 400 variables) to create a separate variable. Recall that you set the **Roll up Terms** property to **Yes** and the **No. of Rolled-up Terms** property to **400**. When you roll up terms, the terms are sorted in descending order of the value of the term weight times the square root of the number of documents. The top 400 highest-ranked terms are then used as variables in the document collection.

18. Close the Text Miner — Interactive window.

19. From the **Model** tab, drag and drop a Decision Tree node into the diagram workspace. Connect the Text Miner — COSTART node to the Decision Tree node. Right-click the Decision Tree node, and select **Rename**. Type *Decision Tree — CT*, where *CT* stands for *COSTART Terms*. Click **OK**.



20. Right-click the Decision Tree — CT node and select **Run**. Click **Yes** in the Confirmation dialog box. Recall that when you created the VAEREXT data set, you set **serious** as the target variable.

21. Click **Results** in the Run Status dialog box after the node has finished running.

22. Select **View** ▸ **Assessment** ▸ **Classification Chart: serious** from the Results window menu to view the Classification Chart.

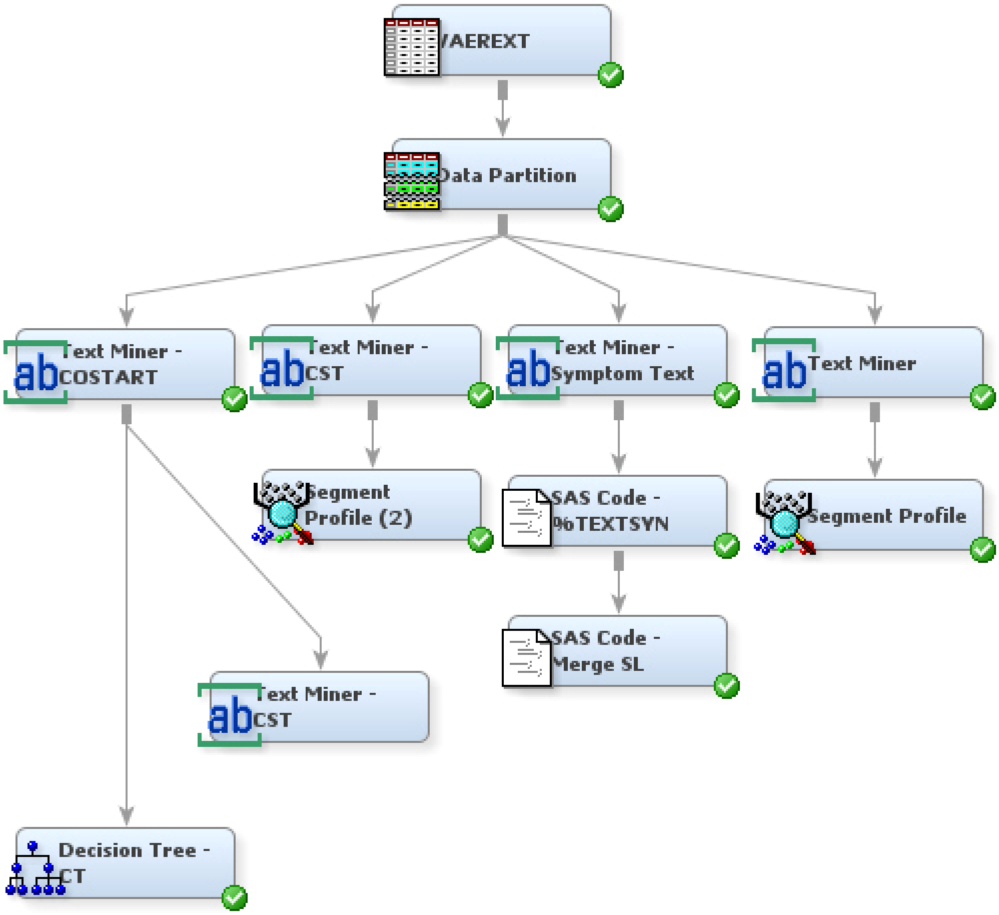
    Note: Blue indicates correct classification and red indicates incorrect classification.

23. Close the Results window.

Use the SYMPTOM_TEXT Variable to Model

To use the SYMPTOM_TEXT variable to create a model, complete the following steps:

1. Right-click the Text Miner — CST node, and select **Copy** from the menu. Right-click an empty space in the diagram workspace and select **Paste**. Connect the Text Miner — COSTART node to the Text Miner — CST node.



This second Text Miner — CST node will be used to analyze the SYMPTOM_TEXT variable. SYMPTOM_TEXT will be the default parse variable because it is the longest text field in the data set. You need to specify COSTRING as a parse variable as well.

2. Select the second Text Miner — CST node. Click the ... button for the **Variables** property in the Properties panel.

3. In the Variables window, set the following:

   - Set the **Use** value of **SYMPTOM_TEXT** to **Yes**.
   - Set the **Use** value of **costring** to **Yes**.
   - Set the **Use** value of **serious** to **Yes**.

   Click **OK**.

4. Set the following properties in the Properties Panel:

   - Set **Compute SVD** to **Yes**.
   - Set **SVD Resolution** to **Low**.
   - Set **Term Weight** to **Mutual Information**.

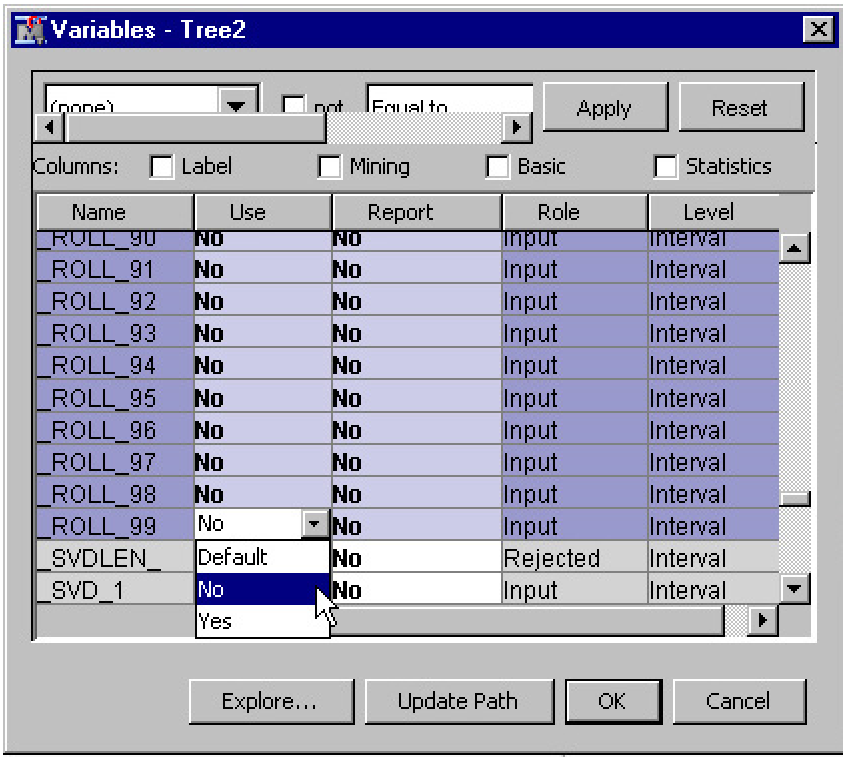
5. Right-click the new Text Miner node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **OK** in the Run Status dialog box.

6. From the **Model** tab, drag and drop a Decision Tree node into the diagram workspace. Connect the new Text Miner node to the Decision Tree node. You will use the decision tree to see whether text mining the original text can do a better job of predicting serious events than just mining the COSTART terms.

7. Right-click the new Decision Tree node and select **Rename**. Type *Decision Tree — ST*, where *ST* stands for *Symptom Text*, in the Node Name text box. Click **OK**.

8. Click the ... button for the **Variables** property in the Decision Tree — ST properties panel. The Variable window opens.

9. Click and scroll to select all of the **_ROLL_** variables, and then set the _ROLL_ **Use** values to **No**.



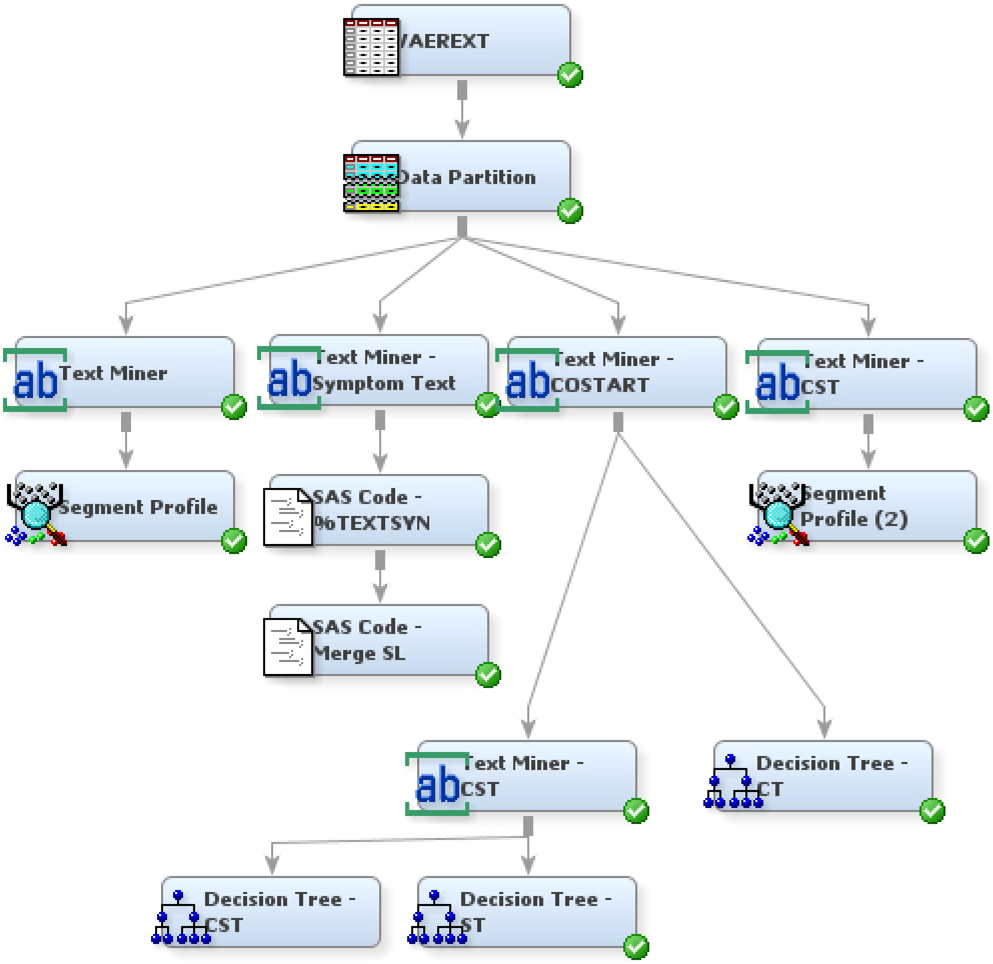
Name	Use	Report	Role	Level
_ROLL_90	No	No	Input	Interval
_ROLL_91	No	No	Input	Interval
_ROLL_92	No	No	Input	Interval
_ROLL_93	No	No	Input	Interval
_ROLL_94	No	No	Input	Interval
_ROLL_95	No	No	Input	Interval
_ROLL_96	No	No	Input	Interval
_ROLL_97	No	No	Input	Interval
_ROLL_98	No	No	Input	Interval
_ROLL_99	No	No	Input	Interval
SVDLEN	Default	No	Rejected	Interval
_SVD_1	No	No	Input	Interval
	Yes			

10. Click **OK** to save your changes.

11. Right-click the Decision Tree — ST node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **OK** in the Run Status dialog box when the node has finished running.

12. From the **Model** tab, drag and drop a Decision Tree node and connect it to the Text Miner — CST node.

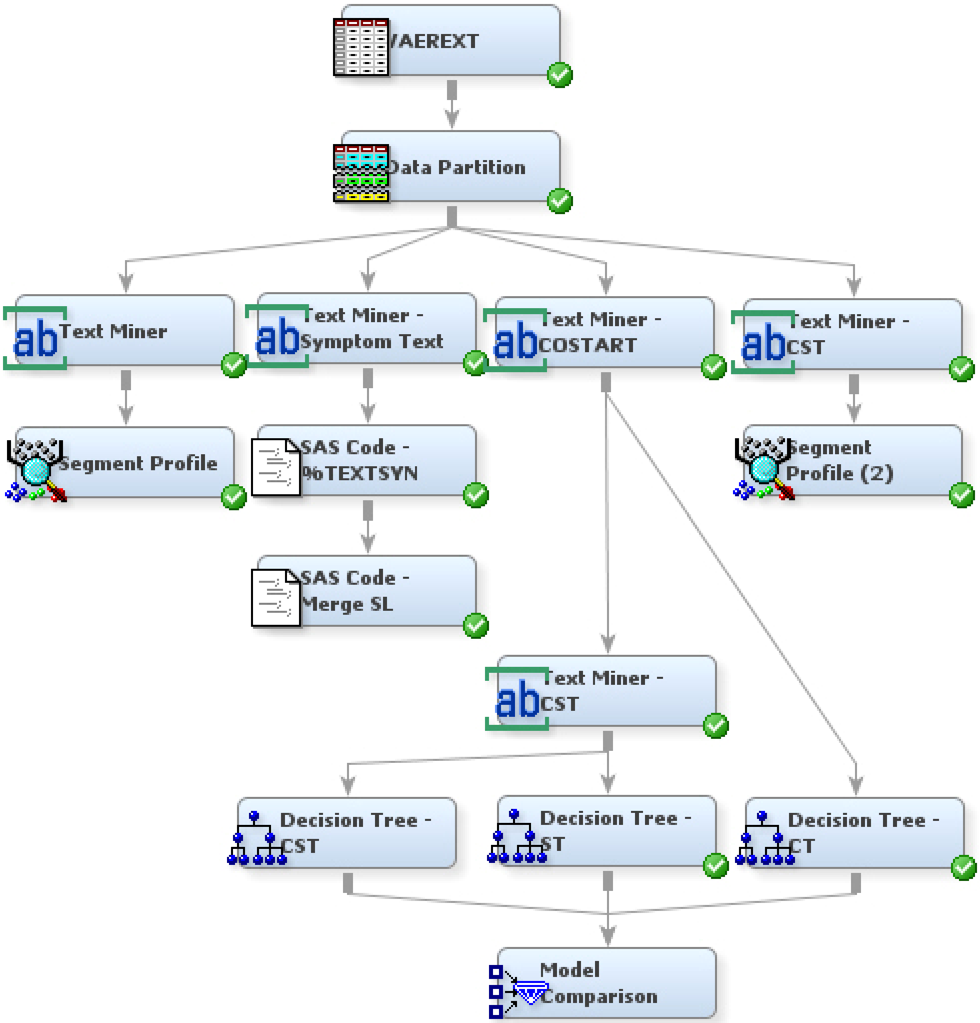
13. Right-click the new Decision Tree node, and select **Rename**. Type *Decision Tree — CST*, where *CST* stands for *COSTART and Symptom Text*, in the Node Name box. Click **OK**. This node will let you see how well you can predict serious events with all the information available to you. Use the default settings for the node.



Compare the Models

To compare the models:

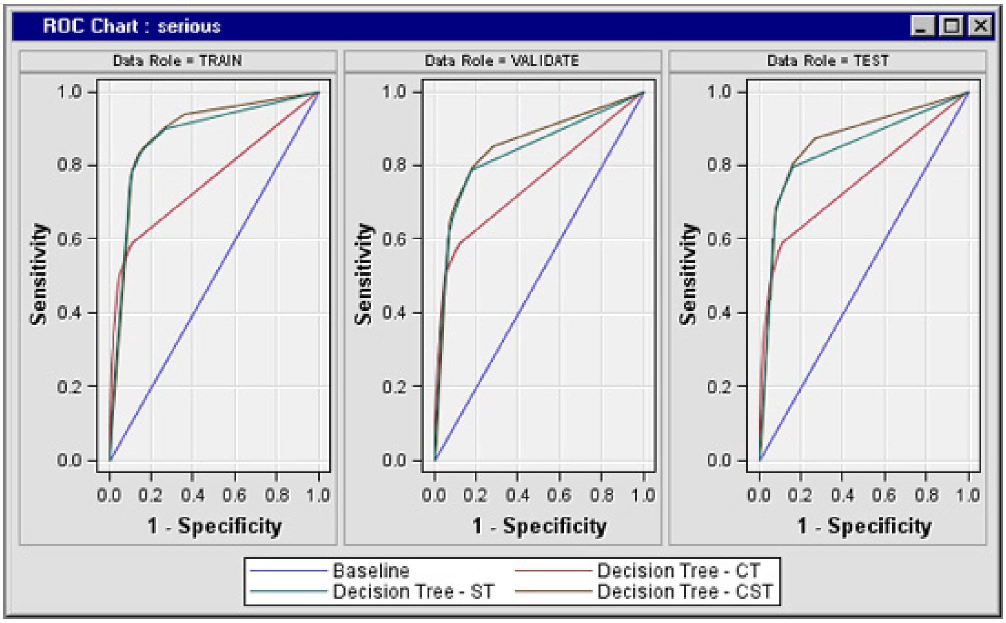
1. From the **Assess** tab, drag and drop a Model Comparison node in the diagram workspace. Connect all three Decision Tree nodes to it. The Model Comparison node enables you to compare the performance of the three different models. Your diagram should look something like the following:



2. Right-click the Model Comparison node, and select **Run**. Click **Yes** in the Confirmation dialog box. Click **Results** in the Run Status dialog box when the Model Comparison node has finished running.

3. Maximize the ROC Chart.

   The greater the area under the curve, the better the model. The red line in the following image shows the results of the model using COSTART terms, the green line shows the results of the SYMPTOM_TEXT terms, and the brown line shows the results of the combined COSTART and SYMPTOM_TEXT terms. The worst model uses only the COSTART terms, while the best model uses a combination of COSTART and SYMPTOM_TEXT. Apparently, text mining can add information not contained in the COSTART terms. The text mining model provides better results than the keyword-based model. Combining the models offers the best results.



4. Select **View** ▸ **Assessment** ▸ **Classification Chart** from the menu at the top of the Results window to view the Classification Chart.

   Note: Blue indicates correct classification and red indicates incorrect classification.

5. Close the Results window. It would be useful to see which variables are most important in the combined model for predicting serious events.

6. Right-click on the Decision Tree — CST node and select **Results** to view the results of the combined Decision Tree models.

7. Click the Output window to maximize it. Scroll through the output to the **Variable Importance** results.

   Note: The SVD terms are more important than the individual terms in predicting a serious adverse event.

8. Minimize the Output window, and then maximize the window that contains the decision tree. Browse the decision tree results.

Additional Exercises

You have looked at predicting the seriousness of adverse events. To explore additional exercises, complete the following steps:

1. You might want to look at the types of adverse events that occur. Try the following:

   - See if you can use the COSTART analysis to predict the clusters you obtained from analyzing the SYMPTOM_TEXT variable. You can do this with the Cluster node. To combine variables together, you might want to try a Decision Tree node.

   - The original data contains other variables, such as medications and lab tests. You know that the type of adverse event is affected by drug interactions. Using the original data, see if you can text mine the medications field to roll up variables for the medications that patients are currently taking. Then use these variables to try to predict the clusters that you obtained for the SYMPTOM_TEXT variable.

2. If you have access to a MedDRA program, run the text through that and perform the same tasks with the MedDRA results that you did with the COSTART terms in this book.

Chapter 7
Tips for Text Mining

Processing a Large Collection of Documents

Using the Text Miner node to process a large collection of documents can require a lot of computing time and resources. If you have limited resources, it might be necessary to take one or more of the following actions:

- Use a sample of the document collection.
- Set some of the parse properties to **No**, such as **Find Entities**, **Noun Groups**, and **Terms in Single Document**.
- Reduce the number of SVD dimensions or roll-up terms. If you are running into memory problems with the SVD approach, you can roll up a certain number of terms, and then the remaining terms are automatically dropped.
- Limit parsing to high information words by turning off all parts of speech other than nouns, proper nouns, noun groups, and verbs.
- Structure sentences properly for best results, including correct grammar, punctuation, and capitalization. Entity extraction does not always generate reasonable results.

Dealing with Long Documents

SAS Text Miner uses the "bag-of-words" approach to represent documents. That means that documents are represented with a vector that contains the frequency with which each term occurs in each document. In addition, word order is ignored. This approach is very effective for short, paragraph-sized documents, but it can cause a harmful loss of information with longer documents. You might want to consider preprocessing your long documents in order to isolate the content that is really of use in your model. For example, if you are analyzing journal papers, you might find that analyzing only the abstract gives the best results. Consider using the SAS DATA step or an alternative programming language such as Perl to extract the relevant content from long documents.

Processing Documents from an Unsupported Language or Encoding

If you have a collection of documents from an unsupported language or encoding, you might still be able to successfully process the text and get useful results. Follow these steps:

1. Set the language to **English**.
2. Turn off these parse properties:
   - **Stem Terms**
   - **Different Parts of Speech**
   - **Noun Groups**
   - **Find Entities**
3. Run the Text Miner node.

Many of the terms might have characters that do not display correctly, but the Interactive Results window should function and you should be able to create stop lists, start lists, and synonym lists.

Chapter 8

Next Steps: A Quick Look at Additional Features

The %TMFILTER Macro

The %TMFILTER macro is a SAS macro that enables you to convert files into SAS data sets. You can use the macro to perform the following tasks:

- Read documents contained in many different formats (such as PDF and Microsoft Word), convert the files to HTML, and create a corresponding SAS data set that can be used as input for the Text Miner node.
- Retrieve Web pages starting from a specified URL and create a SAS data set that can be used as input for the Text Miner node.
- With the language options, separate your collection by language.

Note: The %TMFILTER macro runs only on Windows operating environments.

See **Using the %TMFILTER Macro** in the Text Miner node documentation in SAS Text Miner for more information.

The %TMPUNC Macro

The %TMPUNC macro removes unwanted punctuation from terms in your document collection. If your documents contain terms with run-on punctuation, such as ****people** or **+bags**, these punctuation characters become part of the terms when you parse documents in SAS Text Miner. The %TMPUNC macro enables you to convert terms with run-on punctuation by putting spaces before and after the punctuation characters to prevent them from appearing as part of the term. Without the %TMPUNC macro, these two examples would parse out as two terms, ****people** and **+bags**. After running the %TMPUNC macro, they would parse as five terms:

- *
- *
- **people**
- +

- **bags**

See **Using the %TEXTSYN and %TMPUNC Macros** in the Text Miner node documentation in SAS Text Miner for more information.

Appendix 1
Vaccine Adverse Event Reporting System Data Preprocessing

The VAERS data for 2002-2006 is read into a SAS data set using a SAS program called Vaers_Import.sas. This SAS program creates a table called VAERALL. Vaers_Import.sas is included in the Getting Started with Text Miner 4.1 zip file.

```
proc import out= dmtm9.vaers2006
   datafile= "d:\vaers files\2006vaersdata.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaers2005
   datafile= "d:\vaers files\2005vaersdata.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaers2004
   datafile= "d:\vaers files\2004vaersdata.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaers2003
   datafile= "d:\vaers files\2003vaersdata.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaers2002
   datafile= "d:\vaers files\2002vaersdata.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaervax2006
   datafile= "d:\vaers files\2006vaersvax.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaervax2005
   datafile= "d:\vaers files\2005vaersvax.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaervax2004
   datafile= "d:\vaers files\2004vaersvax.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaervax2003
   datafile= "d:\vaers files\2003vaersvax.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
proc import out= dmtm9.vaervax2002
   datafile= "d:\vaers files\2002vaersvax.csv"
   dbms=csv replace;
   getnames=yes;
   datarow=2;
run;
data dmtm9.vaerall;
   set dmtm9.vaers2002(drop=datedied hospdays)
       dmtm9.vaers2003(drop=datedied hospdays)
       dmtm9.vaers2004(drop=datedied hospdays)
       dmtm9.vaers2005(drop=datedied hospdays)
       dmtm9.vaers2006(drop=datedied hospdays);
run;
data dmtm9.vaervaxall;
   set dmtm9.vaervax2002
       dmtm9.vaervax2003
       dmtm9.vaervax2004
       dmtm9.vaervax2005
       dmtm9.vaervax2006;
run;
```

The data is then further processed to come up with the extract used in the example:

- The separate COSTART terms are appended into a single COSTRING field for each adverse event.
- Additional indicator variables are created for each of the vaccinations received. In the case of DTP, for example, both the Pertussis and Diphtheria/Tetanus variables would be flagged.

The SAS code Vaersetup.sas used to generate the resulting table, VAEREXT, is in the Getting Started with Text Miner 4.1 zip file.

```
libname dmtm9 'd:\emdata\dmtm9';
/*----  TJW Modification: within DATA step  ----*/
%macro FixJunk(TextVar=);
    &TextVar = tranwrd(&TextVar,'n_t ', " not ");
    &TextVar = tranwrd(&TextVar,'N_T ', " NOT ");
    &TextVar = tranwrd(&TextVar,"n't ", " not ");
    &TextVar = tranwrd(&TextVar,"N'T ", " NOT ");
    &TextVar = tranwrd(&TextVar,';', "; ");
    &TextVar = tranwrd(&TextVar,')', " ) ");
    &TextVar = tranwrd(&TextVar,'(', " ( ");
    &TextVar = tranwrd(&TextVar,']', " ] ");
    &TextVar = tranwrd(&TextVar,'[', " [ ");
    &TextVar = tranwrd(&TextVar,'}', " } ");
    &TextVar = tranwrd(&TextVar,'{', " { ");
    &TextVar = tranwrd(&TextVar,'*', " * ");
    &TextVar = tranwrd(&TextVar,',', ", ");
    &TextVar = tranwrd(&TextVar,' w/', " with ");
    *&TextVar = tranwrd(&TextVar,'/', " / ");
    &TextVar = tranwrd(&TextVar,'\', " \ ");
    &TextVar = tranwrd(&TextVar,'~', " ~ ");
    &TextVar = tranwrd(&TextVar,''', " ' ");
    &TextVar = tranwrd(&TextVar,"'s", " ");
    &TextVar = tranwrd(&TextVar,'_', " _ ");
    &TextVar = tranwrd(&TextVar,'&', " and ");
    &TextVar = tranwrd(&TextVar,'.', ".  ");
    &TextVar = tranwrd(&TextVar,'<=', " less than or equal ");
    &TextVar = tranwrd(&TextVar,'>=', " greater than or equal ");
    &TextVar = tranwrd(&TextVar,'<', " less than ");
    &TextVar = tranwrd(&TextVar,'>', " greater than ");
    &TextVar = tranwrd(&TextVar,'=', " equals ");
    &TextVar = trim(left(compbl()));
%mend FixJunk;
```

```
data dmtm9.vaerext(keep=cage_yr sex symptom_text serious numdays pedflag sym_cnt
                  vax_1-vax_16 vax_cnt immun_cnt costring v_adminby v_fundby);
   length coterm $ 25 costring $255;
   array syms{20} $ 25 sym01-sym20;
   array vaxs{8} $ vax1-vax8;
   array nvax{16} vax_1-vax_16;

   set dmtm9.vaerall;

   /* Only include adverse events that occurred within 90 days of vaccination */
   if numdays <= 90;
   if cage_yr = . then cage_yr = 0;
   if cage_mo = . then cage_mo = 0;
   if vax_date ne .;

   /* Serious events are ones that required an overnight hospital stay or caused */
   /*  disability, death, or a life-threatening event                            */
   if l_threat='Y' or died='Y' or hospital='Y' or x_stay='Y' or disable='Y'
   then serious='Y';
   else serious='N';

   /* Determine age of vaccine recipient -- year + month, mark all those under */
   /* 9 as pediatric                                                           */
   cage_yr = cage_yr+cage_mo;
   if cage_yr <=9 then pedflag='Y'; else pedflag='N';
   if died=' ' then died='N';
   if er_visit = ' ' then er_visit='N';
   if recovd = ' ' then recovd='U';

   /* Since serious adverse events are rare (approx 8%) oversample serious events*/
   if serious='N' and uniform(0) < .7 then delete;
```

```
/* Create flag variables for illnesses frequently innoculated against, also*/
/* count up number of immunizations given at one time to a patient as immun_cnt*/
label vax_1='Anthrax'
   vax_2='Diphtheria/Tetanus'
   vax_3='Flu'
   vax_4='Hepatitis A'
   vax_5='Hepatitis B'
   vax_6='HIB (Haemophilus)'
   vax_7='Polio (IPV,OPV)'
   vax_8='Measles,Mumps,Rubella'
   vax_9='Meningoccoccal'
   vax_10='Pneumo (7-valent)'
   vax_11='Pneumo (23-valent)'
   vax_12='Rabies'
   vax_13='Smallpox'
   vax_14='Typhoid'
   vax_15='Pertussis'
   vax_16='Varicella'
   ;

do i=1 to 16;
   nvax{i}=0;
   end;

immun_cnt=0;
do i=1 to min(vax_cnt,8);
   select (vaxs{i});
      when ('6VAX-F') do; vax_2=1; vax_5=1; vax_6=1; vax_7=1;
         immun_cnt=immun_cnt+5; end;
      when ('ANTH') do; vax_1=1; immun_cnt=immun_cnt+1; end;
      when ('DPP') do; vax_2=1; vax_15=1; vax_7=1; immun_cnt=immun_cnt+4; end;
      when ('DT','DTOX','TD','TTOX') do; vax_2=1; immun_cnt=immun_cnt+2; end;
      when ('DTAP','DTP','TDAP') do;
         vax_2=1; vax_15=1; immun_cnt=immun_cnt+3; end;
      when ('DTAPH','DTPHIB') do;
         vax_2=1; vax_15=1; vax_6=1; immun_cnt=immun_cnt+4; end;
      when ('DTAPHE') do;
         vax_2=1; vax_15=1; vax_5=1; vax_7=1; immun_cnt=immun_cnt+5; end;
      when ('FLU','FLUN') do; vax_3=1; immun_cnt=immun_cnt+1; end;
      when ('HBHEPB') do; vax_6=1; vax_5=1; immun_cnt=immun_cnt+2; end;
      when ('HBPV','HBVC','HIBV') do; vax_6=1; immun_cnt=immun_cnt+1; end;
      when ('HEP') do; vax_5=1; immun_cnt=immun_cnt+1; end;
      when ('HEPA') do; vax_4=1; immun_cnt=immun_cnt+1; end;
      when ('HEPAB') do; vax_4=1; vax_5=1; immun_cnt=immun_cnt+2; end;
      when ('IPV','OPV') do; vax_7=1; immun_cnt=immun_cnt+1; end;
      when ('MEA','MER','MM','MMR','MU','MUR','RUB') do;
         vax_8=1; immun_cnt=immun_cnt+3; end;
      when ('MMRV') do; vax_8=1; vax_16=1; end;
      when ('MEN','MNC','MNQ') do; vax_9=1; end;
      when ('PNC') do; vax_10=1; immun_cnt=immun_cnt+1; end;
      when ('PPV') do; vax_11=1; immun_cnt=immun_cnt+1; end;
      when ('RAB','RABA') do; vax_12=1; immun_cnt=immun_cnt+1; end;
      when ('SMALL') do; vax_13=1; immun_cnt=immun_cnt+1; end;
      when ('TYP') do; vax_14=1; immun_cnt=immun_cnt+1; end;
      when ('VARCEL') do; vax_16=1; immun_cnt=immun_cnt+1; end;
```

```
            otherwise;
            end;

         end;
         if immun_cnt > 0;

      /* Create a field, costring, with all the constart terms concatenated */
      /* together in one string                                             */
      costring = '';

      do i=1 to min(sym_cnt,20);
         coterm = syms{i};
         costring=trim(costring) || ' ' || trim(coterm);
         end;

       /* Fix punctuation issues */
       %FixJunk(textvar=symptom_text);

      run;

   proc freq;
      tables pedflag immun_cnt vax_cnt v_adminby v_fundby sex serious vax_1-vax_16;
      run;
```

Glossary

catalog directory
a part of a SAS catalog that stores and maintains information about the name, type, description, and update status of each member of the catalog.

clustering
the process of dividing a data set into mutually exclusive groups so that the observations for each group are as close as possible to one another and different groups are as far as possible from one another. In SAS Text Miner, clustering involves discovering groups of documents that are more similar to each other than they are to the rest of the documents in the collection. When the clusters are determined, examining the words that occur in the cluster reveals the focus of the cluster. Forming clusters within the document collection can help you to understand and summarize the collection without reading every document. The clusters can reveal the central themes and key concepts that are emphasized by the collection.

concept linking
finding and displaying the terms that are highly associated with the selected term in the Terms table.

data source
a data object that represents a SAS data set in the Java-based Enterprise Miner GUI. A data source contains all the metadata for a SAS data set that Enterprise Miner needs in order to use the data set in a data mining process flow diagram. The SAS data set metadata that is required to create an SAS Enterprise data source includes the name and location of the data set; the SAS code that is used to define its library path; and the variable roles, measurement levels, and associated attributes that are used in the data mining process.

diagram
See process flow diagram.

entity
any of several types of information that SAS Text Miner is able to distinguish from general text. For example, SAS Text Miner can identify names (of people, places, companies, or products, for example), addresses (including street addresses, post office addresses, e-mail addresses, and URLs), dates, measurements, currency amounts, and many other types of entities.

libref

a name that is temporarily associated with a SAS library. The complete name of a SAS file consists of two words, separated by a period. The libref, which is the first word, indicates the library. The second word is the name of the specific SAS file. For example, in VLIB.NEWBDAY, the libref VLIB tells SAS which library contains the file NEWBDAY. You assign a libref with a LIBNAME statement or with an operating system command.

model

a formula or algorithm that computes outputs from inputs. A data mining model includes information about the conditional distribution of the target variables, given the input variables.

node

(1) in the SAS Enterprise Miner user interface, a graphical object that represents a data mining task in a process flow diagram. The statistical tools that perform the data mining tasks are called nodes when they are placed on a data mining process flow diagram. Each node performs a mathematical or graphical operation as a component of an analytical and predictive data model. (2) in a neural network, a linear or nonlinear computing element that accepts one or more inputs, computes a function of the inputs, and optionally directs the result to one or more other neurons. Nodes are also known as neurons or units. (3) a leaf in a tree diagram. The terms leaf, node, and segment are closely related and sometimes refer to the same part of a tree.

parsing

to analyze text for the purpose of separating it into its constituent words, phrases, multiword terms, punctuation marks, or other types of information.

partitioning

to divide available data into training, validation, and test data sets.

process flow diagram

a graphical representation of the various data mining tasks that are performed by individual Enterprise Miner nodes during a data mining analysis. A process flow diagram consists of two or more individual nodes that are connected in the order in which the data miner wants the corresponding statistical operations to be performed. Short form: PFD.

roll-up terms

the highest-weighted terms in the document collection.

SAS data set

a file whose contents are in one of the native SAS file formats. There are two types of SAS data sets: SAS data files and SAS data views. SAS data files contain data values in addition to descriptor information that is associated with the data. SAS data views contain only the descriptor information plus other information that is required for retrieving data values from other SAS data sets or from files that are stored in other software vendors' file formats.

scoring

the process of applying a model to new data in order to compute output. Scoring is the last process that is performed in data mining.

segmentation

the process of dividing a population into sub-populations of similar individuals. Segmentation can be done in a supervisory mode (using a target variable and various techniques, including decision trees) or without supervision (using clustering or a Kohonen network).

singular value decomposition

a technique through which high-dimensional data is transformed into lower-dimensional data.

source-level debugger

an interactive environment in SAS that enables you to detect and resolve logical errors in programs that are being developed. The debugger consists of windows and a group of commands.

stemming

the process of finding and returning the root form of a word. For example, the root form of grind, grinds, grinding, and ground is grind.

stop list

a SAS data set that contains a simple collection of low-information or extraneous words that you want to remove from text mining analysis.

test data

currently available data that contains input values and target values that are not used during training, but which instead are used for generalization and model comparisons.

training data

currently available data that contains input values and target values that are used for model training.

validation data

data that is used to validate the suitability of a data model that was developed using training data. Both training data sets and validation data sets contain target variable values. Target variable values in the training data are used to train the model. Target variable values in the validation data set are used to compare the training model's predictions to the known target values, assessing the model's fit before using the model to score new data.

variable

a column in a SAS data set or in a SAS data view. The data values for each variable describe a single characteristic for all observations. Each SAS variable can have the following attributes: name, data type (character or numeric), length, format, informat, and label.

Index

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.

Pharmaceutical Statistics Using SAS: A Practical Guide
Carpenter's Complete Guide to the SAS REPORT Procedure