SAS Publishing

Getting Started with
SAS® 9.1 Text Miner

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *Getting Started with SAS® 9.1 Text Miner*. Cary, NC: SAS Institute Inc.

Getting Started with SAS® 9.1 Text Miner

ISBN 1-59047-353-1

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

1st printing, January 2004

Contents

Chapter 1 △ Introduction to Text Mining and SAS Text Miner 1

What Is Text Mining? **1**

What Is SAS Text Miner? **2**

The Text Mining Process **3**

Tips for Text Mining **3**

Chapter 2 △ Walking through SAS Text Miner with an Example 5

Example Overview **5**

Creating a New Project **6**

Creating the Process Flow Diagram **7**

Viewing the Results **8**

Chapter 3 △ Preparing to Analyze Documents: File Preprocessing 15

What Text Sources Can SAS Text Miner Use? **15**

Using the %tmfilter Macro to Convert Text Files to Data Sets **16**

Chapter 4 △ Beginning the Analysis: Text Parsing 19

What Is Text Parsing? **19**

Examples of Parsed Terms **20**

Using Part-of-Speech and Entity Categories **22**

Stop and Start Lists **23**

Handling Equivalent Terms: Stems and Synonyms **24**

Customizing Equivalent Terms **26**

Chapter 5 △ Producing a Quantitative Representation of the Document Collection 31

From Text to Numbers **31**

Weighting Functions **32**

Transformation (Dimension Reduction) **34**

Chapter 6 △ Exploration and Prediction 37

Overview **37**

Hierarchical Clustering **37**

Expectation-Maximization Clustering **37**

Classification and Prediction **38**

Chapter 7 △ Examples 39

Example 1: Classifying News Articles **39**

Example 2: Scoring New Documents **46**

Appendix 1 △ References 49

References **49**

Appendix 2 △ Recommended Reading 51

Recommended Reading **51**

Index 53

CHAPTER

1

Introduction to Text Mining and SAS Text Miner

What Is Text Mining? 1
What Is SAS Text Miner? 2
The Text Mining Process 3
Tips for Text Mining 3

What Is Text Mining?

The purpose of text mining is to help you understand what the text tells you without having to read every word. Text mining applications fall into two areas: *exploring* the textual data for its content, and then *using* the 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 a call center. 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. The result enables you to better understand the collection.

Predictive mining involves classifying the documents into categories and using the information that is implicit in the text for decision making. You might want to identify the customers who ask standard questions so that they receive an automated answer. Or 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. In data mining terminology, this is known as *predictive modeling*. Predictive modeling involves examining past data to predict future results. You might have a data set that contains information about past buying behaviors, along with comments that the customers made. You can then build a predictive model that can be used to score new customers: that is, in the past these customers did this, so if new customers have similar comments, they are likely to do the same thing. 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 train 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 all the rest.

Both of the above aspects of text mining share some of the same requirements. Namely, text documents that human beings can easily understand must first be represented in a form that can be mined. 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 to a structured form before it can be mined.

What Is SAS Text Miner?

SAS Text Miner contains a sophisticated Text Miner node that can be embedded into a SAS Enterprise Miner process flow diagram. The node analyzes text that exists in a SAS data set, that is in an external database through SAS/ACCESS, or as files in a file system. The Text Miner node encompasses the parsing and exploration aspect of text mining, and sets up the data for predictive mining and further exploration using the rest of the Enterprise Miner nodes. This enables you to analyze the new structured information that you have acquired from the text however you want, combining it with other structured data as desired. The node is highly customizable and allows 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 the documents into meaningful groups and report the concepts that you discover in the clusters. All of this is done 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

□ stemming

□ automatic recognition of multiple-word terms

□ normalization of various entities such as dates, currency, percent, and year

□ part-of-speech tagging

□ extraction of entities such as organizations, products, social security numbers, time, titles, and more

□ support for synonyms.

A secondary tool that Text Miner uses is a SAS macro that is called %tmfilter. This macro accomplishes a text preprocessing step and allows SAS data sets to be created from documents that reside in your file system or on the Web pages. These documents can exist in a number of proprietary formats.

SAS Text Miner is part of Enterprise Miner. Enterprise Miner provides a rich set of data mining tools that facilitate the prediction aspect of text mining. The integration of Text Miner within Enterprise Miner enables the combining of textual data with traditional data-mining variables.

With all of this functionality, SAS Text Miner becomes a very flexible tool that can be used to solve a variety of problems. Below are some examples of tasks that can be accomplished.

□ 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 cost, 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.

Table 1.1 The General Order for Text Mining

Action	Result
File preprocessing	Creates a single SAS data set from your document collection. The SAS data set will be used as input for the Text Mining node. (This is an optional step. Do this if the text is not already in a SAS data set or external database.)
Text parsing	Decomposes textual data and generates a quantitative representation suitable for data mining purposes.
Transformation (dimension reduction)	Transforms the quantitative representation into a compact and informative format.
Document analysis	Performs clustering or classification of the document collection.

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

You might or might not 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.

Tips for Text Mining

Using the Text Miner node to process a very 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.
- □ deselecting some options in the Text Miner Settings window, such as stemming and entity extraction, and the search for words that occur in a single document.
- □ reduce the number of SVD dimensions or roll-up terms. If you have memory problems when you use the SVD approach, you can roll up a certain number of terms, and drop the remaining terms. If you do that and perform SVD at the same time, only the rolled up terms are used in the calculation of SVD. This way you can reduce the size of the problem.
- □ limit parsing to high-information words by deselecting the parts of speech other than nouns, proper nouns, noun groups, and verbs.

CHAPTER 2

Walking through SAS Text Miner with an Example

Example Overview **5**
Creating a New Project **6**
Creating the Process Flow Diagram **7**
Viewing the Results **8**

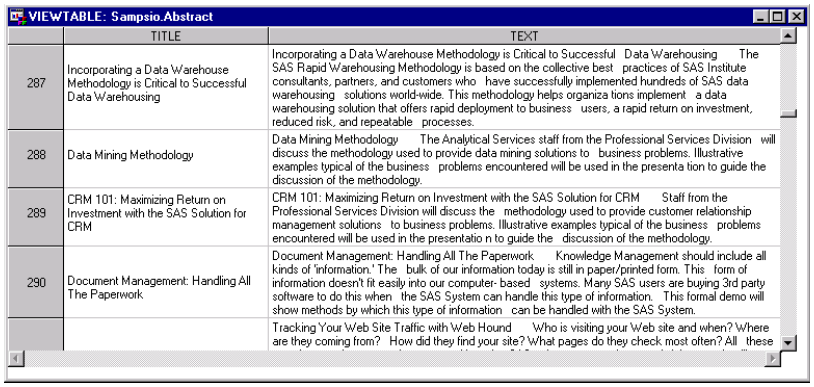
Example Overview

The following example is designed to help you start building a process flow diagram. Several key features are demonstrated:

- □ specifying the input data set
- □ configuring the Text Miner node settings for text parsing and for dimension reduction of the term-by-document frequency matrix
- □ clustering the documents.

Suppose that you work with a large collection of SUGI papers and the goals are to understand what these papers are about and to identify the papers that are related to data warehousing issues. A possible approach is to parse the documents into terms, and then group the documents based on the parsed term through a clustering analysis.

The SAMPSIO.ABSTRACT data set contains information about 1,238 papers that were prepared for meetings of the SAS Users Group International from 1998 through 2001 (SUGI 23 through 26). The following display shows a partial profile of the data set:



VIEWTABLE: Sampsio.Abstract

	TITLE	TEXT
287	Incorporating a Data Warehouse Methodology is Critical to Successful Data Warehousing	Incorporating a Data Warehouse Methodology is Critical to Successful Data Warehousing The SAS Rapid Warehousing Methodology is based on the collective best practices of SAS Institute consultants, partners, and customers who have successfully implemented hundreds of SAS data warehousing solutions world-wide. This methodology helps organiza tions implement a data warehousing solution that offers rapid deployment to business users, a rapid return on investment, reduced risk, and repeatable processes.
288	Data Mining Methodology	Data Mining Methodology The Analytical Services staff from the Professional Services Division will discuss the methodology used to provide data mining solutions to business problems. Illustrative examples typical of the business problems encountered will be used in the presenta tion to guide the discussion of the methodology.
289	CRM 101: Maximizing Return on Investment with the SAS Solution for CRM	CRM 101: Maximizing Return on Investment with the SAS Solution for CRM Staff from the Professional Services Division will discuss the methodology used to provide customer relationship management solutions to business problems. Illustrative examples typical of the business problems encountered will be used in the presentatio n to guide the discussion of the methodology.
290	Document Management: Handling All The Paperwork	Document Management: Handling All The Paperwork Knowledge Management should include all kinds of 'information.' The bulk of our information today is still in paper/printed form. This form of information doesn't fit easily into our computer- based systems. Many SAS users are buying 3rd party software to do this when the SAS System can handle this type of information. This formal demo will show methods by which this type of information can be handled with the SAS System.
		Tracking Your Web Site Traffic with Web Hound Who is visiting your Web site and when? Where are they coming from? How did they find your site? What pages do they check most often? All these

The data set contains two variables:

- □ **TITLE** — is the title of the SUGI paper.
- □ **TEXT** — contains the abstract of the SUGI paper.

Creating a New Project

1 To start Enterprise Miner, you must first have a session of SAS running. You open Enterprise Miner by typing

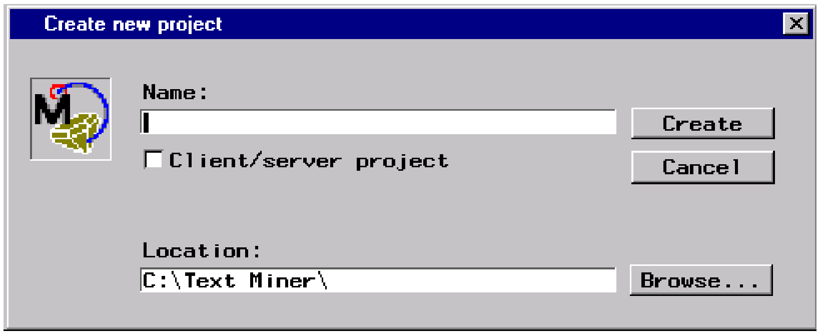
```
miner
```

in the command window in the upper left corner of an open SAS session.

2 Select

New ▶ Project

from the Enterprise Miner main menu. The Create New Project window opens.



Name:

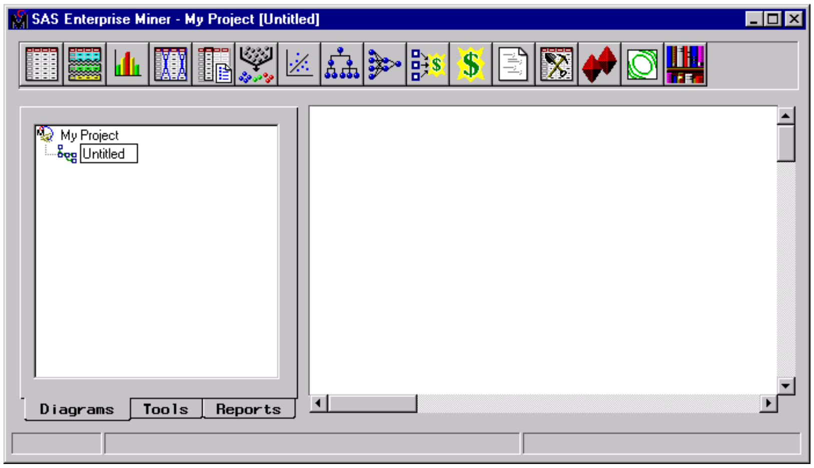
□ Client/server project

Location: C:\Text Miner\

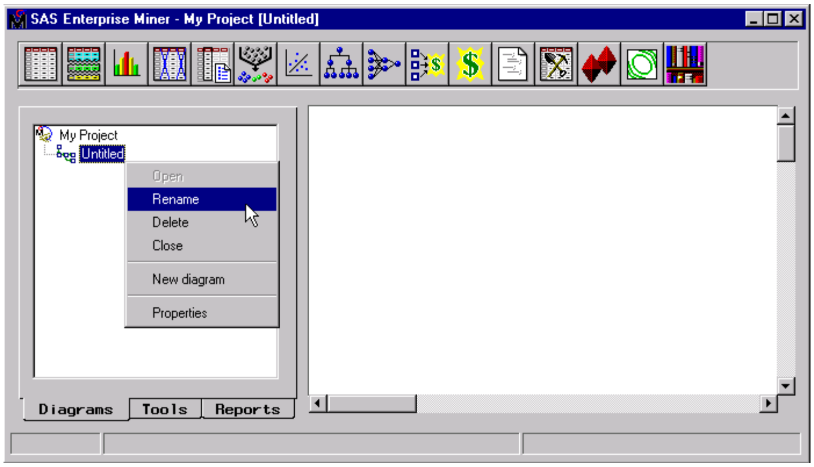
3 Type the project name in the **Name** entry field.

4 A project location will be suggested. You can type a different location for storing the project in the **Location** entry field, or click Browse to search for a location by using a GUI interface.

5 After setting your project location, click Create. Enterprise Miner creates the project that contains a default diagram labeled “Untitled.”

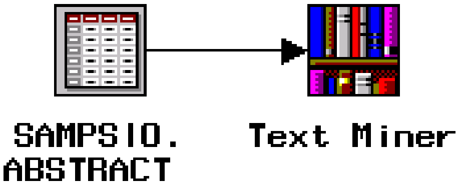


6 To rename the diagram, right-click the diagram icon or label and select **Rename**. Type a new diagram name.



Creating the Process Flow Diagram

Follow these steps to create the Process Flow Diagram:



Input Data Source Node

1 Add an Input Data Source node to the diagram workspace.

2 Open the node and set SAMPSIO.ABSTRACT as the source data set.

3 Select the Variables tab and assign a role of **`input`** to the variable TEXT.

Text Miner Node

1 Add a Text Miner node to the diagram workspace and connect it to the Input Data Source node.

2 Double-click the Text Miner node to open the Text Miner Settings window.

3 Select the Parse tab of the Text Miner Settings window.

- □ Ensure that **`Variable to be parsed`** and **`Language`** are set to TEXT and English, respectively.
- □ Use the **`Identify as Terms`** area to specify the items that are considered as terms in the analysis. In this example, words that occur in a single document, punctuation marks, and numbers are excluded. Ensure that only the **`Same word as different part of speech`**, **`Stemmed words as root form`**, **`Noun groups`**, and **`Entities: Names, Address, etc.`** check boxes are selected.
- □ Use the **`Initial word lists`** area to specify the data sets for a stop or start list, and a synonym list. In this example, use SAMPSIO.SUGISTOP as the stop list. All the words that are in the stop list data set are excluded from the analysis.

4 Select the Transform tab of the Text Miner Settings window.

- □ Select the **`Generate SVD dimensions when running node`** check box and ensure that the value of **`Maximum Dimensions`** is set to 100.
- □ Use the Weight area to specify the weighting methods. In this example, use the default settings.

5 Click OK to save the changes.

6 Click the *Run* tool icon to run the Text Miner node.

Viewing the Results

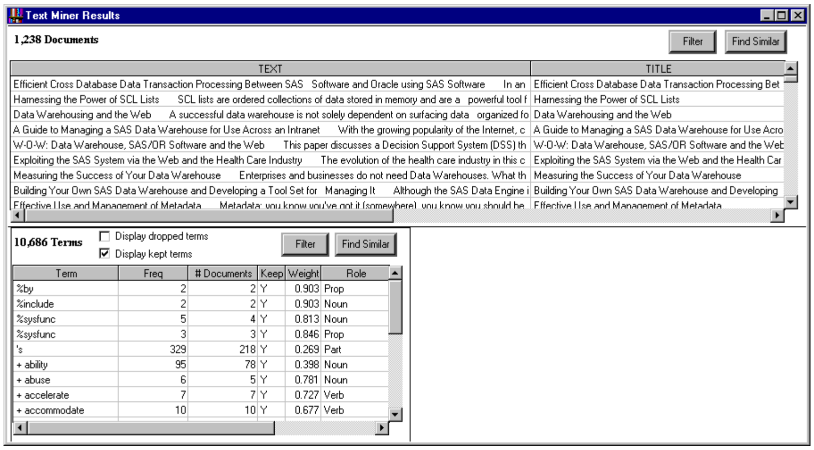
After the node is run successfully, open the Text Miner Results Browser.

1 The Text Miner Results Browser displays two sections:

- □ Documents table
- □ Terms table.

The Documents table displays information about the document collection. In this example, the Documents table displays the abstract and the title of SUGI papers.

The Terms table displays information about all the terms that are found in the document collection. Use the **`Display dropped terms`** and **`Display kept terms`** check boxes to specify the type of terms that are displayed in the Terms table. You also can sort the Terms table by clicking the column heading. In this example, clear the selection of **`Display dropped terms`** and sort the table by the Term column. Following is an example display of the Text Miner Results Browser.



1,238 Documents

TEXT	TITLE
Efficient Cross Database Data Transaction Processing Between SAS Software and Oracle using SAS Software In an	Efficient Cross Database Data Transaction Processing Bet
Harnessing the Power of SCL Lists SCL lists are ordered collections of data stored in memory and are a powerful tool f	Harnessing the Power of SCL Lists
Data Warehousing and the Web A successful data warehouse is not solely dependent on surfacing data organized fo	Data Warehousing and the Web
A Guide to Managing a SAS Data Warehouse for Use Across an Intranet With the growing popularity of the Internet, c	A Guide to Managing a SAS Data Warehouse for Use Acro
W-O-W: Data Warehouse, SAS/OR Software and the Web This paper discusses a Decision Support System (DSS) th	W-O-W: Data Warehouse, SAS/OR Software and the Wet
Exploiting the SAS System via the Web and the Health Care Industry The evolution of the health care industry in this c	Exploiting the SAS System via the Web and the Health Car
Measuring the Success of Your Data Warehouse Enterprises and businesses do not need Data Warehouses. What th	Measuring the Success of Your Data Warehouse
Building Your Own SAS Data Warehouse and Developing a Tool Set for Managing It Although the SAS Data Engine i	Building Your Own SAS Data Warehouse and Developing
Effective Use and Management of Metadata Metadata: you know you've got it (somewhere), you know you should be	Effective Use and Management of Metadata

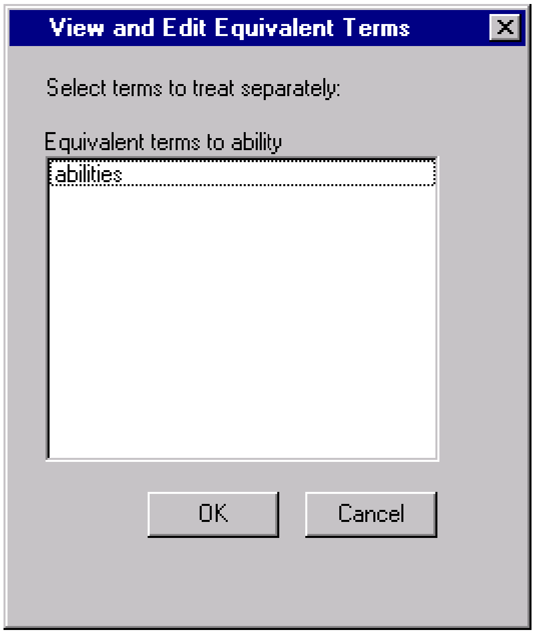
10,686 Terms □ Display dropped terms ☑ Display kept terms

Term	Freq	# Documents	Keep	Weight	Role
%by	2	2	Y	0.903	Prop
%include	2	2	Y	0.903	Noun
%sysfunc	5	4	Y	0.813	Noun
%sysfunc	3	3	Y	0.846	Prop
's	329	218	Y	0.269	Part
+ ability	95	78	Y	0.398	Noun
+ abuse	6	5	Y	0.781	Noun
+ accelerate	7	7	Y	0.727	Verb
+ accommodate	10	10	Y	0.677	Verb

2 In the Terms table, a term that has a plus sign has a group of equivalent terms that you can view. Select the term **`+ ability`**, and from the main menu select

Edit ► View and Edit Equivalent Terms

A window opens to display a list of terms that are equivalent to **`ability`**.



View and Edit Equivalent Terms

Select terms to treat separately:

Equivalent terms to ability

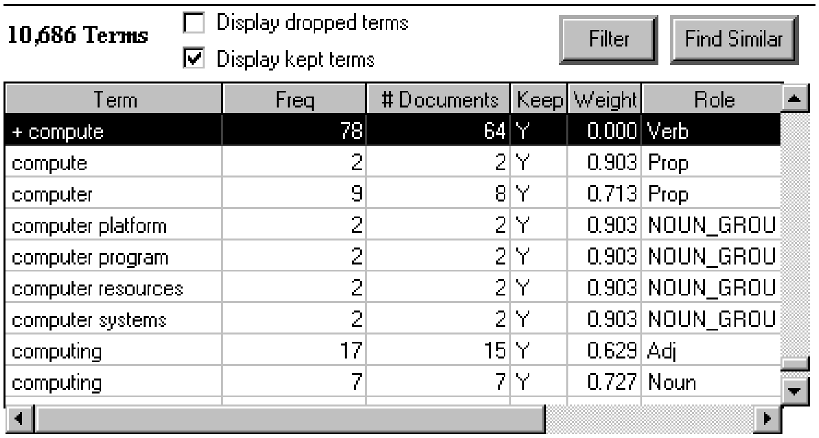
abilities

If you want to drop any of the equivalent terms, select the term and click OK. In this example, click Cancel to return to the Text Miner Results Browser.

3 Examining the parsed terms enables you to find those terms that should be treated the same. In this example, scroll down the Terms table, and you find that the terms **`+ calculate`** and **`+ compute`** have similar meaning and should be treated equivalently. The frequencies of these terms in the document collection are 39 and 39, respectively. You can press CTRL and then click to highlight these terms and select from the main menu

Edit ▶ Treat as Equivalent Terms

The Create Equivalent Terms window prompts you to select the representative term. Select the term **`+ compute`** and click OK. As a result, the term **`+ compute`** and its row might be highlighted and displayed at the top of the Terms table. The frequency of **`+ compute`** is 78, which is the sum of 39 and 39. Note that the weight of the term **`+ compute`** is not really zero. The weight is updated when you request an action that depends on it, such as performing a cluster analysis.



10,686 Terms

☐ Display dropped terms

☑ Display kept terms

Term	Freq	# Documents	Keep	Weight	Role
+ compute	78	64	Y	0.000	Verb
compute	2	2	Y	0.903	Prop
computer	9	8	Y	0.713	Prop
computer platform	2	2	Y	0.903	NOUN_GROU
computer program	2	2	Y	0.903	NOUN_GROU
computer resources	2	2	Y	0.903	NOUN_GROU
computer systems	2	2	Y	0.903	NOUN_GROU
computing	17	15	Y	0.629	Adj
computing	7	7	Y	0.727	Noun

If the term **`+ compute`** is not displayed at the top of the table, right-click in the Terms table and select **`Find`**. Type **`+ compute`** in the **`Find TERM containing`** entry field and click OK.

4 The terms that have a keep status of N are not used in the calculation of SVD dimensions. After examining the Terms table, you might want to eliminate some terms from the analysis such as "+ sas institute" and "sas institute inc." because all the SUGI papers are SAS-related. Select these terms in the Terms table and select from the main menu

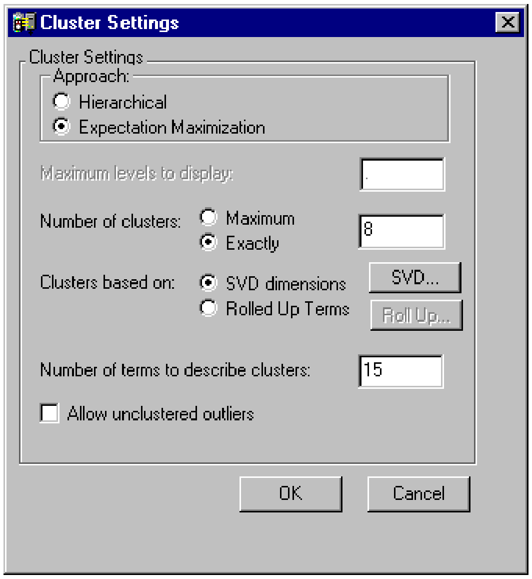
Edit ▶ Toggle Keep Status

to change the keep status of these terms from Y to N.

5 Now, you have finished making changes to the terms. The next step is to group the documents by applying a clustering analysis. Select from the main menu

Tools ▶ Cluster

to open the Cluster Settings window.



Cluster Settings

Cluster Settings

Approach:

○ Hierarchical

● Expectation Maximization

Maximum levels to display:

Number of clusters: ○ Maximum ● Exactly 8

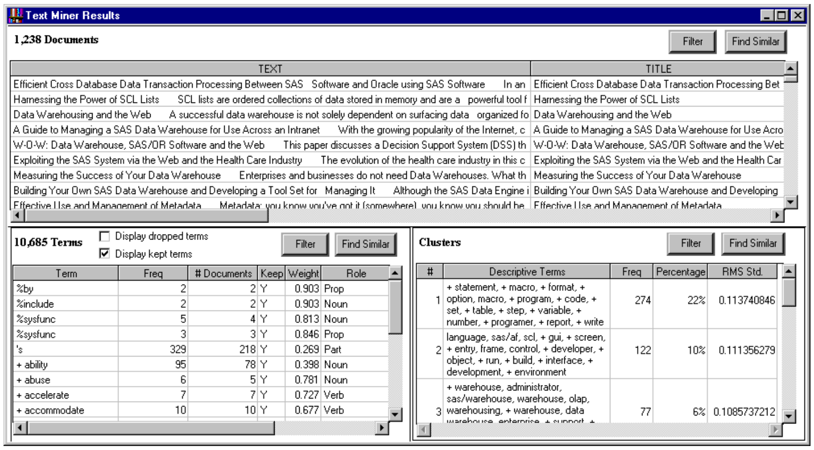
Clusters based on: ● SVD dimensions ○ Rolled Up Terms

Number of terms to describe clusters: 15

□ Allow unclustered outliers

In the Cluster Settings window, you specify the clustering method to use, either hierarchical clustering or expectation-maximization clustering, and the inputs for the clustering analysis, either SVD dimensions or roll-up terms. In this example, use the SVD dimensions as inputs for the expectation-maximization clustering analysis, select **`Exact`** rather than **`Maximum`**, and change the number of clusters to 8. Also, change the number of terms to describe clusters to 15. Click OK to generate the clusters.

6 The Clusters table displays the descriptive terms for each of the eight clusters. Also, the Documents table displays another variable, Cluster ID, which represents the cluster that a document is grouped into.



Text Miner Results

1,238 Documents

TEXT	TITLE
Efficient Cross Database Data Transaction Processing Between SAS Software and Oracle using SAS Software In an	Efficient Cross Database Data Transaction Processing Bet
Harnessing the Power of SCL Lists SCL lists are ordered collections of data stored in memory and are a powerful tool f	Harnessing the Power of SCL Lists
Data Warehousing and the Web A successful data warehouse is not solely dependent on surfacing data organized fo	Data Warehousing and the Web
A Guide to Managing a SAS Data Warehouse for Use Across an Intranet With the growing popularity of the Internet, c	A Guide to Managing a SAS Data Warehouse for Use Acro
W-O-W: Data Warehouse, SAS/OR Software and the Web This paper discusses a Decision Support System (DSS) th	W-O-W: Data Warehouse, SAS/OR Software and the Web
Exploiting the SAS System via the Web and the Health Care Industry The evolution of the health care industry in this c	Exploiting the SAS System via the Web and the Health Car
Measuring the Success of Your Data Warehouse Enterprises and businesses do not need Data Warehouses. What th	Measuring the Success of Your Data Warehouse
Building Your Own SAS Data Warehouse and Developing a Tool Set for Managing It Although the SAS Data Engine i	Building Your Own SAS Data Warehouse and Developing
Effective Use and Management of Metadata Metadata: you know you've got it (somewhere), you know you should be	Effective Use and Management of Metadata

10,685 Terms

☐ Display dropped terms
☑ Display kept terms

Term	Freq	# Documents	Keep	Weight	Role
%by	2	2	Y	0.903	Prop
%include	2	2	Y	0.903	Noun
%sysfunc	5	4	Y	0.813	Noun
%sysfunc	3	3	Y	0.846	Prop
's	329	218	Y	0.269	Part
+ ability	95	78	Y	0.398	Noun
+ abuse	6	5	Y	0.781	Noun
+ accelerate	7	7	Y	0.727	Verb
+ accommodate	10	10	Y	0.677	Verb

Clusters

#	Descriptive Terms	Freq	Percentage	RMS Std.
1	+ statement, + macro, + format, + option, macro, + program, + code, + set, + table, + step, + variable, + number, + programer, + report, + write	274	22%	0.113740846
2	language, sas/af, scl, + gui, + screen, + entry, frame, control, + developer, + object, + run, + build, + interface, + development, + environment	122	10%	0.111356279
3	+ warehouse, administrator, sas/warehouse, warehouse, olap, warehousing, + warehouse, data warehouse, enterprise, + support, +	77	6%	0.1085737212

In this example, all the documents are grouped into ten clusters. By examining the descriptive terms in each cluster, you see the following clusters in the collection of SUGI abstracts.

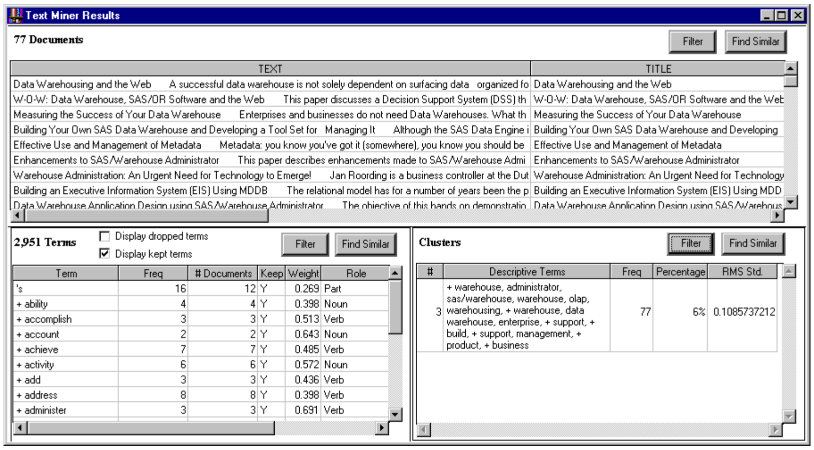
Table 2.1 Clusters Extracted from the Abstracts

Descriptive Terms	Cluster
+ statement, + macro, + format, + option, macro, + program, + code, + set, + table, + step, + variable, + number, + programer, + report, + write	programming
language, sas/af, scl, + gui, + screen, + entry, frame, control, + developer, + object, + run, + build, + interface, + development, + environment	SAS/AF issues
+ warehouse, administrator, sas/warehouse, warehouse, olap, warehousing, + warehouse, data warehouse, enterprise, + support, + build, + support, management, + product, + business	data warehousing issues
delivery, + technology, + organization, + year, + business, management, + development, + solution, 's, + problem, + develop, + good, + process, + provide, + report	spurious cluster
sas/intrnet, + output, html, web, + output, + graph, output, delivery, + page, + create, sas/graph, web, + browser, ods, version	output issues
+ business, + customer, + decision, + relationship, + process, + problem, + solution, between, management, + approach, such as, + set, used to, + technique, through	CRM issues
windows, + server, + feature, nt, + client, version, java, server, + performance, + platform, windows, + interface, + version, + technology, + cover	architecture issues
+ compute, + analysis, + compare, regression, + study, + response, statistical, + model, + test, + procedure, + method, standard, + design, between, + result	statistical analysis

Notice that some of the clusters contain terms that might need to be excluded from the analysis. You can add these terms to the stop list, refresh samples in the Input Data Source node, and rerun the clustering analysis.

7 Recall that one of the goals is to identify the SUGI papers that are related to data warehousing issues. By examining the descriptive terms in each of the clusters,

you find that the third cluster is related to data warehousing issues. To display the documents in that cluster and the corresponding terms, select the third cluster and click the corresponding Filter button. The Documents table displays the abstract and title of the SUGI papers that are in the cluster of data warehousing issues.



77 Documents

TEXT	TITLE
Data Warehousing and the Web A successful data warehouse is not solely dependent on surfacing data organized fo	Data Warehousing and the Web
W-O-W: Data Warehouse, SAS/OR Software and the Web This paper discusses a Decision Support System (DSS) th	W-O-W: Data Warehouse, SAS/OR Software and the Web
Measuring the Success of Your Data Warehouse Enterprises and businesses do not need Data Warehouses. What th	Measuring the Success of Your Data Warehouse
Building Your Own SAS Data Warehouse and Developing a Tool Set for Managing It Although the SAS Data Engine i	Building Your Own SAS Data Warehouse and Developing
Effective Use and Management of Metadata Metadata: you know you've got it (somewhere), you know you should be	Effective Use and Management of Metadata
Enhancements to SAS/Warehouse Administrator This paper describes enhancements made to SAS/Warehouse Admi	Enhancements to SAS/Warehouse Administrator
Warehouse Administration: An Urgent Need for Technology to Emerge! Jan Roording is a business controller at the Dut	Warehouse Administration: An Urgent Need for Technology
Building an Executive Information System (EIS) Using MDDB The relational model has for a number of years been the p	Building an Executive Information System (EIS) Using MDD
Data Warehouse Application Design using SAS/Warehouse Administrator The objective of this hands on demonstratio	Data Warehouse Application Design using SAS/Warehous

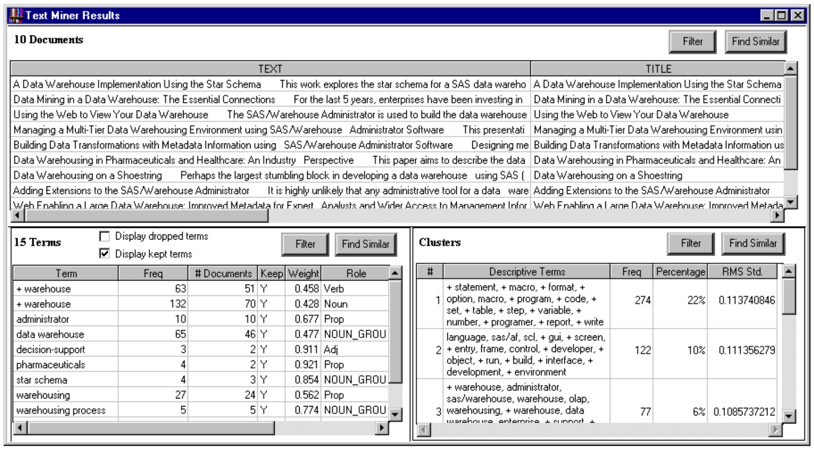
2,951 Terms

Term	Freq	# Documents	Keep	Weight	Role
's	16	12	Y	0.269	Part
+ ability	4	4	Y	0.398	Noun
+ accomplish	3	3	Y	0.513	Verb
+ account	2	2	Y	0.643	Noun
+ achieve	7	7	Y	0.485	Verb
+ activity	6	6	Y	0.572	Noun
+ add	3	3	Y	0.436	Verb
+ address	8	8	Y	0.398	Verb
+ administer	3	3	Y	0.691	Verb

#	Descriptive Terms	Freq	Percentage	RMS Std.
3	+ warehouse, administrator, sas/warehouse, warehouse, olap, warehousing, + warehouse, data warehouse, enterprise, + support, + build, + support, management, + product, + business	77	6%	0.1085737212

Select the *Show All* tool icon from the toolbox to return to the original display.

8 You might want to find the *n* most similar documents or terms with respect to the term **`+ warehouse`**. To do this, find the noun **`+ warehouse`** in the Terms table by right-clicking in the Terms table and selecting **`Find`**. Type **`+ warehouse`** in the **`Find TERM containing`** entry field and click OK. Then, select the term **`+ warehouse`** and click Find Similar to search for the similar items. The following example displays the ten closest documents with respect to the term **`+ warehouse`**.



10 Documents

TEXT	TITLE
A Data Warehouse Implementation Using the Star Schema This work explores the star schema for a SAS data wareho	A Data Warehouse Implementation Using the Star Schema
Data Mining in a Data Warehouse: The Essential Connections For the last 5 years, enterprises have been investing in	Data Mining in a Data Warehouse: The Essential Connecti
Using the Web to View Your Data Warehouse The SAS/Warehouse Administrator is used to build the data warehouse	Using the Web to View Your Data Warehouse
Managing a Multi-Tier Data Warehousing Environment using SAS/Warehouse Administrator Software This presentati	Managing a Multi-Tier Data Warehousing Environment usin
Building Data Transformations with Metadata Information using SAS/Warehouse Administrator Software Designing me	Building Data Transformations with Metadata Information us
Data Warehousing in Pharmaceuticals and Healthcare: An Industry Perspective This paper aims to describe the data	Data Warehousing in Pharmaceuticals and Healthcare: An
Data Warehousing on a Shoestring Perhaps the largest stumbling block in developing a data warehouse using SAS (	Data Warehousing on a Shoestring
Adding Extensions to the SAS/Warehouse Administrator It is highly unlikely that any administrative tool for a data ware	Adding Extensions to the SAS/Warehouse Administrator
Web Enabling a Large Data Warehouse: Improved Metadata for Expert Analysts and Wider Access to Management Infor	Web Enabling a Large Data Warehouse: Improved Metada

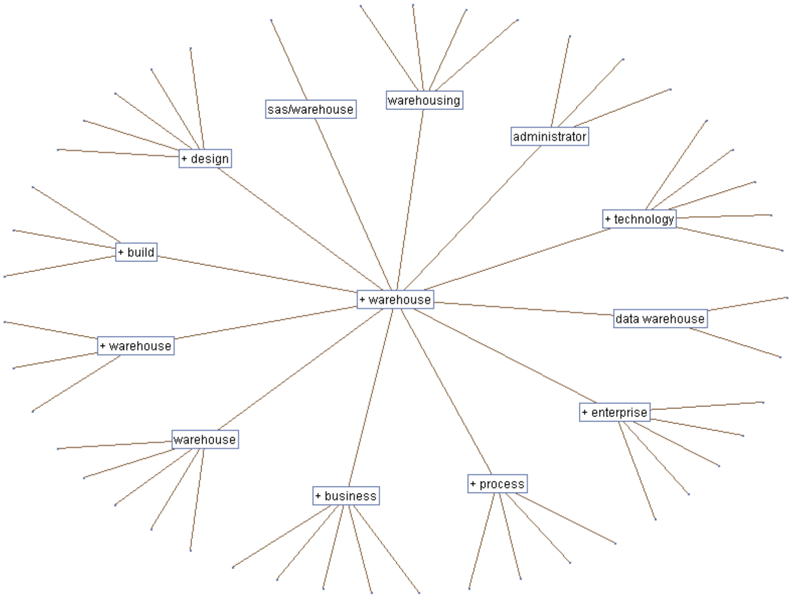
15 Terms

Term	Freq	# Documents	Keep	Weight	Role
+ warehouse	63	51	Y	0.458	Verb
+ warehouse	132	70	Y	0.428	Noun
administrator	10	10	Y	0.677	Prop
data warehouse	65	46	Y	0.477	NOUN_GROU
decision-support	3	2	Y	0.911	Adj
pharmaceuticals	4	2	Y	0.921	Prop
star schema	4	3	Y	0.854	NOUN_GROU
warehousing	27	24	Y	0.562	Prop
warehousing process	5	5	Y	0.774	NOUN_GROU

#	Descriptive Terms	Freq	Percentage	RMS Std.
1	+ statement, + macro, + format, + option, macro, + program, + code, + set, + table, + step, + variable, + number, + programer, + report, + write	274	22%	0.113740846
2	language, sas/af, scl, + gui, + screen, + entry, frame, control, + developer, + object, + run, + build, + interface, + development, + environment	122	10%	0.111356279
3	+ warehouse, administrator, sas/warehouse, warehouse, olap, warehousing, + warehouse, data warehouse, enterprise, + support, +	77	6%	0.1085737212

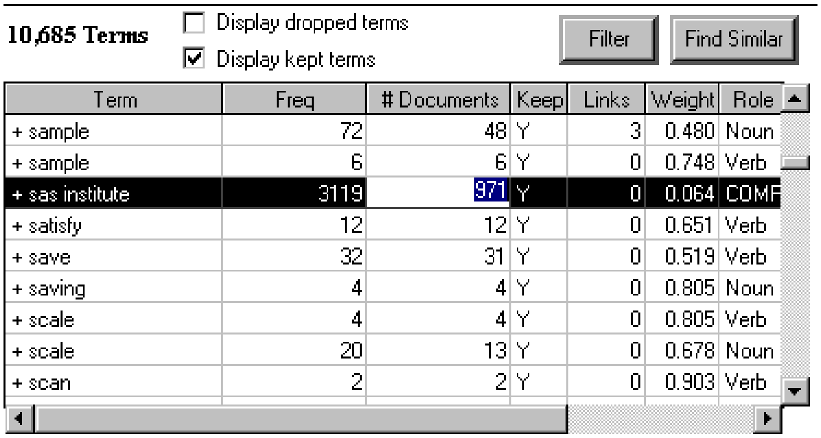
Select the *Show All* tool icon from the toolbox to return to the original display.

9 Find the noun **`+ warehouse`** and select it. Then, select **`View Concept Links`** from the pop-up menu. The Web browser opens and displays the linkage between **`+ warehouse`** and other terms.



By examining these terms that commonly occur together, you can modify the Terms table. For example, set the nouns **`+ warehouse`** and **`data warehouse`** to be equivalent terms, and re-cluster the documents. Close the Web browser.

Notice that the Terms table now has a column **`Links`**, which indicates the number of concept links. The following display shows the Links column.


10,685 Terms □ Display dropped terms ☑ Display kept terms

Term	Freq	# Documents	Keep	Links	Weight	Role
+ sample	72	48	Y	3	0.480	Noun
+ sample	6	6	Y	0	0.748	Verb
+ sas institute	3119	971	Y	0	0.064	COMF
+ satisfy	12	12	Y	0	0.651	Verb
+ save	32	31	Y	0	0.519	Verb
+ saving	4	4	Y	0	0.805	Noun
+ scale	4	4	Y	0	0.805	Verb
+ scale	20	13	Y	0	0.678	Noun
+ scan	2	2	Y	0	0.903	Verb

If the term that you select has no links, a dialog window will inform you that no concept links are found for this term.

CHAPTER 3

Preparing to Analyze Documents: File Preprocessing

What Text Sources Can SAS Text Miner Use? **15**
Using the %tmfilter Macro to Convert Text Files to Data Sets **16**

What Text Sources Can SAS Text Miner Use?

SAS Text Miner can support textual data wherever it resides: inside a SAS table, as a character object that is stored in an external database, or in a file system.

The Text Miner node assumes that the document is represented as a variable in a SAS data set, or as a character or a character large object in an external database that can be accessed by a SAS/ACCESS engine. Other variables in the data set might include structured information that can then be combined with the textual data to improve your mining process.

If your data is stored in external files or on the Web, you can use the SAS macro %tmfilter to create a SAS data set. The SAS data set that %tmfilter generates can contain the entire text or a portion of each document (TEXT), a path to the original documents (URI), a path to the HTML version of the documents that is used for parsing (FILTERED), and other variables. If your documents are longer than 32KB, the TEXT variable stores a portion of the text and the documents are referenced with a URL. Another strategy is to break the document into smaller pieces.

These external files can be any mix of a variety of formats, including Adobe PDF, ASCII, HTML, and Microsoft Word. The following table lists the supported document formats:

Table 3.1 Supported Document Formats

Document Format	Version
Adobe Portable Document Format (PDF)	1.1 to 4.0
Applix Asterix	Applix Asterix
Applix Spread Sheet	10
ASCII text	ASCII text
Corel Presentations	7.0, 8.0
Corel Quattro Pro for Windows	7.0, 8.0
Document Content Architecture (DCA)-RTF	sc23-0758-1
Framemaker Interchange Format (MIF)	5.5
HTML	All

Document Format	Version
IBM DisplayWrite	1.0, 1.1
Lotus 1-2-3	2, 3, 4, 96, 97, R9
Lotus AMI pro	2.0, 3.0
Lotus Word pro	96, 97, R9
Microsoft Excel	3, 4, 5, 97, 98, 2000
Microsoft PowerPoint	4.0, 95, 97
Microsoft Rich Text Format	All
Microsoft Word	1.x, 2.0, 6.0, 7.0, 8.0, 95, 97, 2000
Microsoft Word for DOS	2.2 to 5.0
Microsoft Word for MAC	4.x, 5.x, 6.x, 98
Microsoft Works	1.0, 2.0, 3.0, 4.0
WordPerfect for DOS	5.0, 6.0
WordPerfect for MAC	2.2, 3.0
WordPerfect for Windows	7.0
XYWrite	4.12

Using the %tmfilter Macro to Convert Text Files to Data Sets

You use the %tmfilter macro to filter the text out of documents in various formats, or to retrieve Web pages by starting from a specified URL. %tmfilter generates a SAS data set that can be used as input for the Text Miner node.

The general form of the macro for converting files of various formats into SAS data sets is

```
%tmfilter(dataset=data-set-name,dir=path-to-original-documents,
destdir=path-to-HTML-files, numchar=n)
```

where *data-set-name* refers to the name of the SAS data set that the macro generates, *path-to-original-documents* is the path to the directory that contains the files to be filtered by %tmfilter, and *path-to-HTML-files* is the path to the filtered HTML files. The attribute *destdir* is optional if you want to contain the text in the data set rather than store it in the file system. *Numchar* is an optional attribute that specifies the number of characters that will be stored in the TEXT variable. The default is 60. Note that unlike the prior version of Text Miner, this variable is NOT parsed, but is provided as a summary to aid the viewer in identifying the document. The macro automatically processes all files of the appropriate formats in that folder and all its subfolders.

The general form of the macro for retrieving Web pages from a given URL and creating SAS data sets is

```
%tmfilter(dataset=data-set-name,dir=path-to-original-documents,
destdir=path-to-HTML-files, url=URL, depth=integer)
```

where *URL* is the Uniform Resource Locator and *integer* is the number of levels of the URL to process.

In addition, you can use the option language=*list-of-languages* in %tmfilter to automatically identify the language of the documents.

Several additional attributes are available in %tmfilter. See Using the %tmfilter Macro section in the Text Miner node documentation for more information.

CHAPTER 4

Beginning the Analysis: Text Parsing

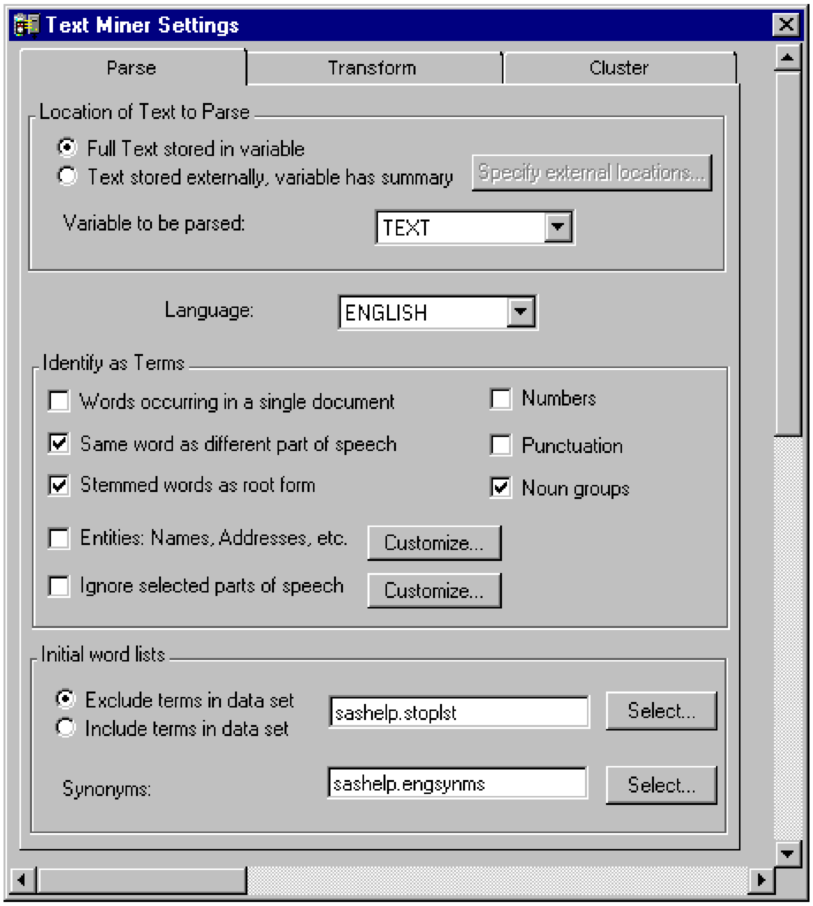
What Is Text Parsing? **19**
Examples of Parsed Terms **20**
Handling Multiword Terms and Phrases **21**
Handling Entities **21**
Handling Foreign Languages **22**
Using Part-of-Speech and Entity Categories **22**
Part-of-Speech Categories **22**
Entity Categories **23**
Stop and Start Lists **23**
Handling Equivalent Terms: Stems and Synonyms **24**
Automatic Stemming **24**
Synonyms **25**
Treating Entities of Canonical Forms **25**
Customizing Equivalent Terms **26**
Synonyms and Part of Speech **27**
Synonyms, Entities, and Noun Groups **29**

What Is Text Parsing?

Text parsing is the first step in analyzing your document collection. Parsing enables you to

- break sentences or documents into terms
- extract particular entities that are meaningful to your specific application
- find the root form (stem) of a word and specify synonyms
- remove low-information words such as *a*, *an*, and *the*.
- identify the term’s part of speech
- create a quantitative representation for the collection of documents.

The following display shows the available parsing options in the Text Miner node:



Location of Text to Parse

● Full Text stored in variable

○ Text stored externally, variable has summary

Variable to be parsed: TEXT

Language: ENGLISH

Identify as Terms

☐ Words occurring in a single document

☑ Same word as different part of speech

☑ Stemmed words as root form

☐ Entities: Names, Addresses, etc.

☐ Ignore selected parts of speech

☐ Numbers

☐ Punctuation

☑ Noun groups

Initial word lists

● Exclude terms in data set

○ Include terms in data set

sashelp.stoplst

Synonyms: sashelp.engsynms

Examples of Parsed Terms

In SAS Text Miner, text parsing breaks text into components beyond the level of words. For example, parsed terms might optionally include words, phrases, multiword terms, entities, numbers, and punctuation marks. The following table shows examples of how sentences are parsed when the **Same word as different part of speech**, **Stemmed words as root form**, **Entities: Names, Addresses, etc.**, **Numbers**, and **Noun Groups** check boxes are selected.

Table 4.1 Examples of Parsed Sentences

Sentence	Parsed Terms
Coca-Cola announced earnings on Saturday, Dec. 12, 2000. Profits were up by 3.1% as of 12/12/1999.	coca-cola + announce earnings on saturday dec. 12 2000 + profit + be up by 3.1% as of 2000–12–12 1999-12-12
Douglas Ivester earned $12.8 million.	douglas (Location) douglas (Prop) ivester douglas ivestor + earn $12800000 usd

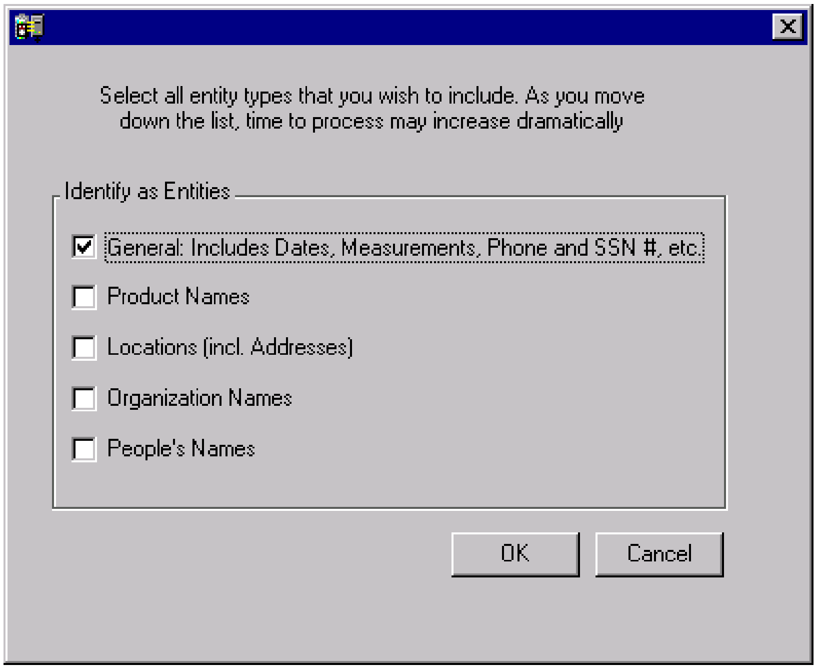
Handling Multiword Terms and Phrases

A multiword term is a group of words to be processed as a single term. A multiword term is typically an idiomatic phrase, a collection of noun-noun or adjective-noun compounds, or a proper name. Following are some examples:

- due to
- because of
- Web browser
- capital goods
- inclement weather

Handling Entities

Your document collection often contains entities of special interest such as names, addresses, companies, and measurements. The Text Miner node can identify and extract these particular entities when the text is parsed. Note that when an entity is parsed, as in the table above, both the entity and its component parts are represented as terms. For example, December can be recognized separately as a month in addition to being recognized inside a date. The following display shows the available entity types in the Text Miner node:



Select all entity types that you wish to include. As you move down the list, time to process may increase dramatically

Identify as Entities

☑ General: Includes Dates, Measurements, Phone and SSN #, etc.

☐ Product Names

☐ Locations (incl. Addresses)

☐ Organization Names

☐ People's Names

Handling Foreign Languages

SAS Text Miner supports several languages: English, Danish, Dutch, Finnish, French, German, Italian, Portuguese, Spanish, and Swedish, although noun group extraction is available only for English, French, German, and Spanish, and entity extraction is available only for English, French, and German.

Using Part-of-Speech and Entity Categories

If the **`Same word as different parts of speech`** check box is selected, each parsed term is assigned a part-of-speech category, unless the parsed term is an entity as described in the previous section. An entity term is given an entity category. This enables you to remove terms from the analysis based on their categories (part of speech or entity) and to emphasize certain terms by increasing the weighting based on the category.

Part-of-Speech Categories

SAS Text Miner identifies the part of speech for each term based on its context. It looks up each term and internally lists all of its possible parts of speech. It then traces all possible choices for the string and analyzes the probable part of speech for each term, based on a set of sequence probabilities. As a result, each term is assigned to a grammatical category based on its role in the given sentence. For more information, see the Part-of-Speech Categories table in SAS Text Miner Appendixes, which are available by selecting from the Enterprise Miner main menu

Help ▶ EM Reference ▶ SAS Text Miner ▶ SAS Text Miner Appendixes

Entity Categories

If a parsed term is an entity, it will be assigned an entity category. Following is the list of entity categories that are available in SAS Text Miner:

- address
- company
- currency
- date
- internet
- location
- measure
- organization
- percent
- person
- phone
- product
- social security number (SSN)
- time
- time_period
- title

For more information about the entity categories, see the Entity Categories table in SAS Text Miner Appendixes, which are available from the Enterprise Miner main menu.

Help ▶ EM Reference ▶ SAS Text Miner ▶ SAS Text Miner Appendixes

Stop and Start Lists

Not all words in the text carry important information. For example, words such as prepositions, articles, and conjunctions have little information in the context of a sentence. Such low-information words can be removed during text parsing since they often do not add useful information to the analysis.

The Text Miner node enables you to use a stop list to control which words are removed from the analysis. The stop list is a simple collection of low-information or extraneous words that you want to ignore during processing. The SASHELP directory contains a default stop list for the following supported language.

- English — sashelp.stoplst
- French — sashelp.frchstop
- German — sashelp.grmnstop

The stop lists are useful for some, but not all, text mining tasks. You can copy the stop lists and edit the copy by adding and removing terms as necessary. One useful approach is to use the default stop list, and then examine the text parsing results. By default, the terms are sorted by descending number of documents in the collection. If you see words that occur in a large percentage of documents in the entire collection, you might want to add some of these words to the stop list and apply changes that are based on your judgment or domain knowledge. In other situations, some of the words in this stop list may not be informative for general text, but within the given domain, the words might have a specialized meaning. In this case, you can remove those words from the stop list so that they, too, would be kept in the analysis.

In contrast to the stop list, you can use a start list to control which words are included in your analysis. SAS Text Miner does not provide a default start list. The start list enables you to examine only certain words of interest. All other terms are removed from the results. This capability might be useful when you have previously mined data in this domain and have customized it. You can then save a start list from the results, and use it in future runs with similar data.

See the Contents of Default Stop Lists tables in SAS Text Miner Appendixes in the Help for information about contents of the stop lists. The appendixes are available by selecting

Help ▶ EM Reference ▶ SAS Text Miner ▶ SAS Text Miner Appendixes

from the Enterprise Miner main menu.

Handling Equivalent Terms: Stems and Synonyms

Automatic Stemming

Stemming means finding and returning the root form (or base form) of a word. Stemming enables you to work with linguistic forms that are more abstract than those of the original text. For example, the stem of *grind*, *grinds*, *grinding*, and *ground* is *grind*. Of course, *ground* can also be a noun whose meaning is completely unrelated to the term *grind* and SAS Text Miner handles this appropriately too. You can configure the Text Miner node settings to treat the same words that have different part of speech as separate terms.

To better understand the advantages of analyzing more abstract terms, suppose that you are grouping documents according to the key words that they contain. But if this is done without stemming, words such as grind, grinds, grinding, and ground will be handled as unrelated words. Documents that contain one of these variants will not be treated the same as documents that contain the other variants. If stemming is applied, all of the documents are grouped under the root form of the verb *grind*. The following table shows more examples of stems.

Table 4.2 Examples of Stemming

Stem	Terms
aller (French)	vais, vas, va, allons, allez, vont
reach	reaches, reached, reaching
big	bigger, biggest
balloon	balloons
go	goes

Stemming usually, but not always, generates better predictive models when you perform stemming and SVD. But stemming is especially critical when you roll up terms.

Synonyms

The document collection often contains terms that do not have the same base form but share the same meaning in context. In SAS Text Miner, you use a synonym list to store groups of equivalent terms. The synonym list consists of records and each record contains the root term, an equivalent term, and the entity category. The way that SAS Text Miner handles a term and its equivalent terms is equivalent to how stems and their roots are treated. A synonym list can be applied to other words that should be treated equivalently but are not direct stems. For example, the words *teach*, *instruct*, *educate*, and *train* do not have a common stem, but share the same meaning of *teach*. In this case, you might want to add terms such as *instruct*, *instructing*, *educate*, *train*, and *training* into the synonym list (mapping to *teach*) in order for them to be also treated the same as *teach*.

Treating Entities of Canonical Forms

Entities can also have synonyms. These are referred to as *canonical forms*. The Text Miner node enables you to specify variant names of particular entities to be treated the same as their canonical form. *Canonical form* is the standard name of a particular entity, such as company name, date, currency, and percentage expressions. For example, IBM and International Business Machines are variant names for the same company. Typically, the longest and most precise version of a person name or organization name within a single document is used as the canonical form.

Date, currency, and percent expressions have the standard ISO (International Standards Organization) format as their canonical forms.

□ date and year — YYYY-MM-DD, YYYY, YYYY-MM, or MM-DD.

□ currency — a number followed by a space and an ISO three-character current code (USD, PTE, FRF, NLG, SEK, ROL, GBP, ESP, RUR, INR, ATS, and KRW). The German comma is replaced by a decimal point, and periods that are used to delimit sets of three decimal places are removed.

□ percentage — a number followed by a percentage. The number can include a minus sign and decimal points, but commas are removed. The German comma is replaced by a decimal point, and periods that are used to delimit sets of three decimal places are removed.

The following table shows examples of how date, currency, and precentage expressions are normalized in SAS Text Miner.

Examples of expressions in textual documents	Terms in parsing results
Date and Year	
3/15/97	1997-03-15
the 22nd of May	—05-22
March 9th, 1961	1961-03-09
’99	1999
Currency	
$100	100 usd
twelve new pence	.12 GBP
£5 GBP	5 GBP
25 cts	.25 usd

Examples of expressions in textual documents	Terms in parsing results
Percent	
21%	21%
thirteen percentage points	13%
seventeen pc.	17%
twenty-seven hundred percent	2700%

Customizing Equivalent Terms

To customize how SAS Text Miner handles the equivalent terms, you must create a synonym list and save it as a SAS data set. This data set must contain three variables that are named term, parent, and category. The following table shows an example of a synonym list:

Table 4.3 Synonym List

Term	Parent	Category
appearing	appear	verb
sasv9	v9	
EM	SAS Enterprise Miner	PRODUCT
administrative assistant	employee	NOUN_GROUP
employees	employees	noun
employeee	employees	noun

The variable Term contains the parsed term. The variable Parent represents the base form, canonical form, or the synonym of the parsed term. The variable Category contains the part of speech, or the entity category if the parsed term is an entity.

- □ The entry for “appearing” shows the use of a synonym list for stemming. All the occurrences of “appearing” are stemmed to the base form “appear.” The category (or role) is relevant only if the **Same word as different part of speech** check box is selected. Otherwise, the part of speech is ignored and all occurrences of “appearing” are stemmed to “appear.”
- □ The entry for “sasv9” shows the use of “v9” as a synonym for “sasv9,” regardless of the part of speech that is assigned.
- □ The entry for “EM” shows a canonical form of the entity extraction component in the software. In this case, it adds a new product into the entities dictionary. If either “EM” or “SAS Enterprise Miner” are found in the text, they are grouped in the canonical form “SAS Enterprise Miner.”
- □ The entry for “administrative assistant” shows the addition of a synonym for a multiword term. If the parsed term in the term column contains more than one word, this entry are added to the multiword dictionary. In this example, all the occurrences of “administrative assistant” in the category NOUN_GROUP are grouped with the term “employee.”

- □ The entry for “employees” shows the use of a synonym list to make unavailable the stemming for a given word while keeping the stemming of other words. In this example, all the occurrences of “employees” are stemmed to “employees” rather than to “employee.”
- □ The entry for “employeee” shows the use of a synonym list to handle misspelled words. Suppose you know that the term “employee” has been misspelled as “employeee” in some documents. When you use a synonym list, all the occurrences of “employeee” will be stemmed to “employees” instead of being treated as a single term.

To treat a multiword term such as data mining as a single term, add the following entry to your synonym list.

Term	Parent	Category
data mining		

Terms in the synonym list are not case sensitive. The synonym will be applied to all instances of the term regardless of case. That is, if the term “father” is in the synonym list with the parent “dad,” then occurrences in the text of “Father” and “FATHER” are returned as “father” and these terms are assigned the parent “dad.”

If a term in the synonym list does not have an assigned category, every occurrence of the term will be stemmed to the parent.

Synonyms and Part of Speech

Selecting the **Same word as different part of speech** check box in the Text Miner Settings window affects how terms in the synonym list are stemmed to their parent (if any). When the **Same word as different part of speech** check box is selected, a term is distinguished by the term and category pair. The following list summarizes how the check box works.

- □ In the following example, a term and the parent, but not the category, are defined.

Term	Parent	Category
well	water	

When the **Same word as different part of speech** check box is selected, every occurrence of “well,” regardless of the part of speech, is assigned the parent “water.”

Similarly, when the **Same word as different part of speech** check box is not selected, all occurrences of “well” are stemmed to “water.”

- □ In the following example, a term and the category, but not the parent, are defined.

Term	Parent	Category
well		noun

When the **Same word as different part of speech** check box is selected, the synonym list that has only a single-word term has no effect on the results. However, if you change the term to a multi-word term such as “data mining,” all occurrences of “data mining” as a noun are identified as a single term.

By contrast, when the **Same word as different part of speech** check box is not selected, the synonym list that has only a single-word term has no effect on the results. If you change the term to a multiword term such as “data mining,” all occurrences of “data mining” (including noun and verb) are identified as a single term.

□ In the following example, a term, the parent, and the category are defined.

Term	Parent	Category
well	water	noun

When the **Same word as different part of speech** check box is selected, the occurrences of “well” as a noun are stemmed to “water.”

When the **Same word as different part of speech** check box is not selected, all occurrences of “well” are stemmed to “water.”

Consider the following synonym list.

Term	Parent	Category
well	water	Noun
well	good	Adj

Term	Parent	Category
Well	Water	Noun
well	good	Noun

If you use the first synonym list when the **Same word as different part of speech** check box is not selected, the first entry for the term “well” (well water Noun) in the synonym list will be used. All the occurrence of “well” are stemmed to “water.” If you use the second synonym list when the **Same word as different part of speech** is selected, the first entry (well water Noun) in the synonym list will be used for text parsing.

In addition, when the **Same word as different part of speech** check box is selected, you cannot assign synonym to a term that has a different part of speech from the actual term. For example, consider the following sentences.

I want to travel the roads of North Carolina.

His commute took him two hours to get to work every day.

You cannot assign “commute” (noun) to have the parent “travel” (verb) because these terms are different parts of speech. The following synonym lists are valid. In the first example, when “commute” occurs as a noun, it will have the parent “travel.” In the second example when “travel” occurs as a verb, it will have the parent “commute.”

Term	Parent	Category
commute	travel	Noun

Term	Parent	Category
travel	commute	Verb

Synonyms, Entities, and Noun Groups

If the term in the synonym list is an entity or noun group, the term will not be stemmed to the parent that you specified. For example, suppose you have the following synonym list.

Term	Parent	Category
North Carolina	home	

The term “North Carolina” is an entity (LOCATION), which is detected by SAS Text Miner. In this case, “North Carolina” is displayed as a single term, which is separated from the term “home.” The term “North Carolina” has a role of LOCATION in the text mining results.

But if you modify the entry in the synonym list by adding a value for Category, then the occurrences of “North Carolina” will be stemmed to “home.”

Term	Parent	Category
North Carolina	home	LOCATION

CHAPTER 5 Producing a Quantitative Representation of the Document Collection

From Text to Numbers **31**
Weighting Functions **32**
Transformation (Dimension Reduction) **34**
Roll-Up Terms **34**
Singular Value Decomposition **34**

From Text to Numbers

One of the goals of text parsing is to create a quantitative representation for the documents. That is, in the process of text parsing, a term-by-document frequency matrix is created. Each entry in the matrix represents the number of times that a term appears in a document.

Suppose that you have a document collection as given below. Documents 1, 3, and 6 are about banking at a financial institution. To be more specific, documents 3 and 6 are about borrowing from a financial institution. Documents 2, 4, 5, and 7 are about the bank of a river. Finally, documents 8 and 9 are about a parade. Some of these documents share the same words. “Bank” can relate to a financial institution or to the shore of a river. “Check” can serve as a noun in document 1 or in an entirely different role as a verb in document 8. “Floats” is used as both a verb in documents 4 and 7, and as an object that appears in a parade in document 8.

- □ Document 1 — Deposit the cash and check in the bank.
- □ Document 2 — The river boat is on the bank.
- □ Document 3 — borrow based on credit
- □ Document 4 — A river boat floats up the river.
- □ Document 5 — A boat is by the dock near the bank.
- □ Document 6 — With credit, I can borrow cash from the bank.
- □ Document 7 — Boat floats by the dock near the river bank.
- □ Document 8 — Check the parade route to see the floats.
- □ Document 9 — along the parade route

Parsing this document collection generates the following term-by-document frequency matrix:

Table 5.1 Term-by-Document Frequency Matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9
the	2	2	0	1	2	1	2	2	1
cash	1	0	0	0	0	1	0	0	0
check	1	0	0	0	0	0	0	1	0
bank	1	1	0	0	1	1	1	0	0
river	0	1	0	2	0	0	1	0	0
boat	0	1	0	1	1	0	1	0	0
+ be	0	1	0	0	1	0	0	0	0
on	0	1	1	0	0	0	0	0	0
borrow	0	0	1	0	0	1	0	0	0
credit	0	0	1	0	0	1	0	0	0
+ float	0	0	0	1	0	0	1	1	0
by	0	0	0	0	1	0	1	0	0
dock	0	0	0	0	1	0	1	0	0
near	0	0	0	0	1	0	1	0	0
parade	0	0	0	0	0	0	0	1	1
route	0	0	0	0	0	0	0	1	1
parade route	0	0	0	0	0	0	0	1	1

The columns in the table correspond to documents. The rows represent parsed terms that include stems and canonical forms. The entry gives the number of times that a term occurs in a document. Some of the words (for example, *and* and *in*) do not appear in the table because they are removed by a stop list. The word *deposit* does not appear in the table because it appears in a single document in the collection.

In this table, a document is represented by the number of times that each of the different terms occurs for that document, or as a 17–dimensional column vector.

This term-by-document frequency matrix serves as the foundation for analysis of the document collection. Sometimes performance can be improved by adjusting the entries with various weighting functions. Often the functions are based on how often a term occurs in the document collection as a whole. The terms that occur relatively infrequently have the highest weights because the subset of documents that include them are typically quite alike.

Weighting Functions

Entries in the term-by-document frequency matrix are generally weighted before they are used in further analysis. The total weight of an entry is determined by its frequency weight and term weight.

Frequency weights are functions of the term frequency alone. The following frequency weights are available:

- □ Binary — Every frequency becomes either a zero or a 1. Binary weight is generally used for documents that contain a very small vocabulary, such as a collection of computer programs that are all written in the same programming language.
- □ Log — The log (base 2) of every frequency plus one is taken. Log weight lessens the effect of a single word being repeated often. This is the default. It is the most generally useful, because a word that appears ten times is probably slightly more significant than a word that appears only once or twice.
- □ None — The frequencies are used directly, with no weights.

Term weights consider the word counts in the document collection. The following term weights are available:

- □ *Entropy* applies the highest weights to terms that occur infrequently in the document collection. This weighting method emphasizes words that occur in few documents within the collection.
- □ *Inverse Document Frequency* uses the reciprocal of the number of documents that a term appears in the collection as the weight, for a similar result to the entropy method. This weighting method emphasizes terms that occur only in few documents within the collection.
- □ *Global Frequency Times Inverse Document Frequency* magnifies the inverse document frequency by multiplying it by the global frequency. This method weights frequent terms more severely than Entropy or Inverse Document Frequency.
- □ *Normal* is the proportion of times that the word occurs in the document collection.
- □ *None* sets all weights to 1.

The Chi-Squared, Mutual Information, and Information Gain weighting methods are based on the term's frequency and its relationship to some other target variable.

- □ *Chi-Squared* uses the value of the chi-squared test statistic for a one-way table of the term and the target.
- □ *Mutual Information* indicates how closely the distribution of documents that contain the term matches the distribution of documents that are contained in each of the target values.
- □ *Information Gain* indicates the expected reduction in entropy that is caused by partitioning the collection by that term. It indicates how well the term or the absence of that term predicts the category.

Mutual Information and Information Gain weighting methods are related to the Entropy measure. These methods are used in the field of information theory.

Term importance weightings are used to help distinguish some terms as being more important than others are. The general guideline is that terms that are useful in categorizing documents are those that occur frequently, but only in a few documents. The documents that contain those terms will be easy to set apart from the rest of the collection. The two weightings that perform similarly in this regard are *entropy* and *inverse document frequency*. The *inverse document frequency* is referred to as tfidf in some cases. In the information retrieval and text mining research, using one of these two weightings gives the best performance.

Term weighting has the effect of magnifying the importance of certain words while reducing the importance of others. The complete matrix generally contains thousands of terms that are used throughout the document collection, of which only a small subset is contained in any one document. Therefore, computing with the entire set of terms is prohibitively expensive. Furthermore, processing high dimensional data is inherently difficult for modeling. Reducing the dimension will improve performance.

Transformation (Dimension Reduction)

Roll-Up Terms

Roll-up terms use the highest weighted terms in the document collection and have variables for each document for the weighted frequencies of each of these terms. It is most useful when the documents are short, so there is little word overlap within the documents. Otherwise, the SVD approach is usually preferable.

Roll-up terms represent each document as a subset of terms that are found across all documents. The method eliminates most of the terms in the collection by using only the *n* highest weighted terms as variables. The parameter *n* is chosen by the user. For any given document, the variable that is associated with a given term contains the weighted frequency of the term. If the particular term does not exist in the document, then its value is zero for that document.

Roll-up terms then use the co-occurrence of terms from the matrix as a measure of similarity between documents. Consider the matrix in Table 5.1. You can see that, by this measure, documents 1 and 2 are more similar than documents 1 and 3. The reason for this is that documents 1 and 2 share the same word "bank," while documents 1 and 3 have no words in common. However, in fact, documents 1 and 2 are not related at all, but document 1 and 3 are similar. So, although the approach is simple, it is not always effective. The reason is that language contains many words that mean the same thing but are written differently as well as words that are written the same but mean something different.

When you are building a predictive model, it is often but not always preferable to use a target-based weighting when you roll up terms. The chi-square term weight technique often gives the best results.

The following section covers the next type of dimension reduction technique that is implemented in SAS Text Miner and that is preferred for most situations: singular value decomposition.

Singular Value Decomposition

The singular value decomposition is an important dimension reduction technique that has been used in the field of information retrieval for years. Its benefits have only recently been applied to text mining.

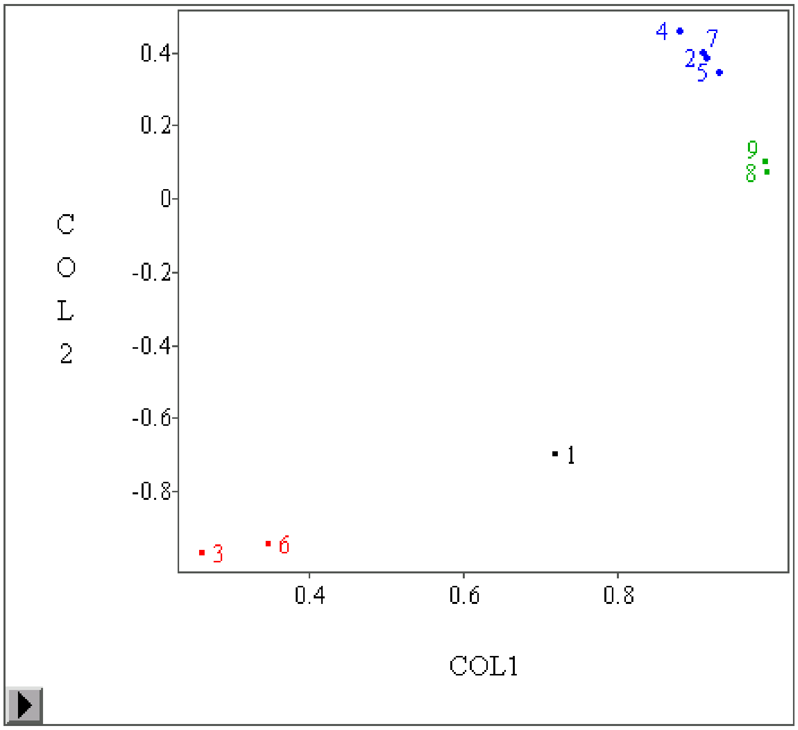
The singular value decomposition determines the "best least square fit" to the original weighted frequency matrix, given a certain number of dimensions, *k*. A higher value of *k* gives a better approximation to the original matrix. However, choosing too large a value for *k* might result in too high a dimension for the modeling process. Generally, the value of *k* must be large enough to preserve the meaning of the document collection, but not so large that it captures the noise. Values between 10 and several hundred are appropriate unless the document collection is small. In SAS Text Miner, you can specify the maximum number of dimensions (*k*) to be calculated for the SVD. As a general rule, smaller values of *k* (2 to 50) are useful for clustering, and larger values (30 to 200) are useful for prediction or classification.

For more information about singular value decomposition, see Singular Value Decomposition in the SAS Text Miner Appendixes, which are available from the Enterprise Miner main menu.

Help ► EM Reference ► SAS Text Miner ► SAS Text Miner Appendixes

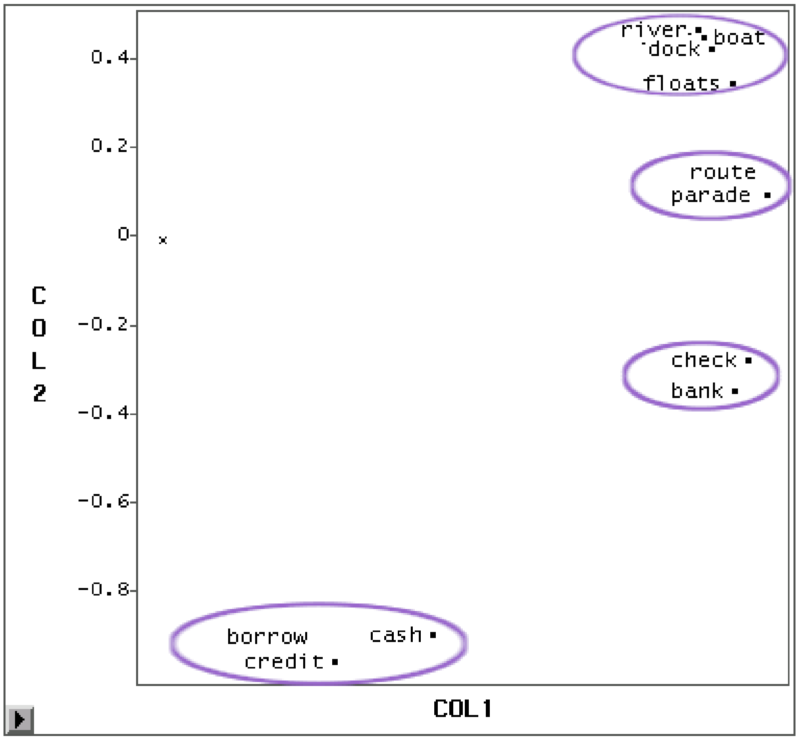
The following description shows the benefits of the singular value decomposition.

If the singular value decomposition is applied to the matrix that is given in Table 5.1, the document is projected into a reduced dimensional space. The generated SVD dimensions are those that fit the subspace the best in terms of least-square best fit. The following display shows the two-dimensional scatter plot of documents.

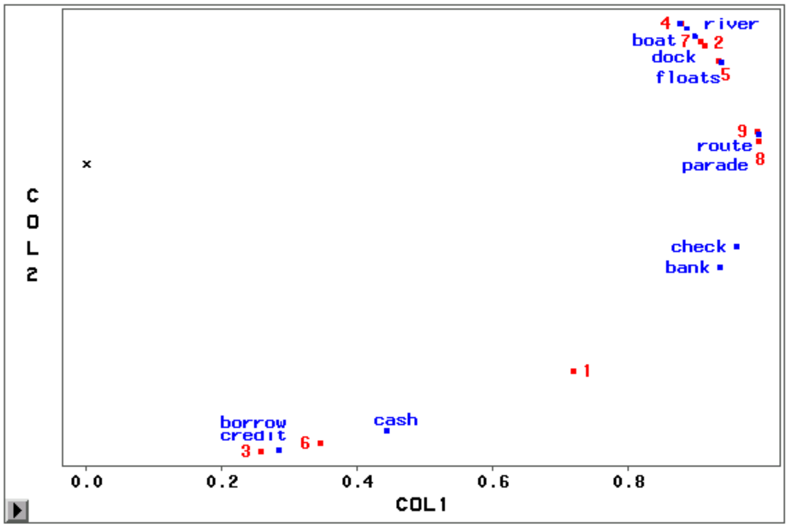


Document 1 is now closer to document 3 than it is to document 2. This is true even though documents 1 and 3 do not share any of the same words. On the other hand, document 5 is directly related to documents 2, 4, and 5. That is, projections tend to place similar documents, even if they share few common words, close to one another in the reduced space. The singular value decomposition represents terms with 2 dimensions rather than the original 11 dimensions (1 dimension for each word).

The following display shows the two–dimensional scatter plot of terms. The terms form four groups.



The following display shows the scatter plot of documents and terms all together.



You should have at least a hundred documents in the document collection in order to get a meaningful result. If you have fewer, the calculation of SVD may fail to converge or to compute the full number of dimensions that were specified.

CHAPTER

6

Exploration and Prediction

Overview **37**
Hierarchical Clustering **37**
Expectation-Maximization Clustering **37**
Classification and Prediction **38**

Overview

Almost every goal of text mining falls into one of three categories: clustering (sometimes called unsupervised learning), classification (or supervised learning), or prediction (classification is actually a form of prediction). Using the parsed terms that occur in documents, clustering involves finding 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 its focus. SAS Text Miner provides two clustering methods: hierarchical clustering and expectation-maximization clustering.

Hierarchical Clustering

Hierarchical clusters are organized so that one cluster may be entirely contained within another cluster, but no other kind of overlap between clusters is allowed. What emerges is a document taxonomy. SAS Text Miner uses the CLUSTER procedure with Ward's minimum-variance method to generate hierarchical clusters. For more information about PROC CLUSTER and Ward's method, see the SAS/STAT documentation.

Expectation-Maximization Clustering

Expectation-maximization clustering performs observation clustering by identifying the primary and secondary clusters. Primary clusters are the densest regions of data points, while secondary clusters are typically less dense groups of those data points that are not summarized in the primary clusters. The expectation-maximization clustering technique is a spatial clustering technique, but, unlike the more common k-means algorithm, expectation-maximization clustering allows clusters to be of arbitrary size and shape.

For more information about the expectation-maximization algorithm, see Expectation-Maximization Clustering in the SAS Text Miner Appendixes, which are available by selecting from the Enterprise Miner main menu

Help ► EM Reference ► SAS Text Miner ► SAS Text Miner Appendixes

.

Enterprise Miner also has other nodes, Clustering and SOM/Kohonen nodes, that are suitable for clustering analysis. 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. Applications include discovering the content of a large knowledge base and analyzing the contents of e-mail, customer comments, abstracts, and survey data. Another application of clustering is to cluster the words of the collection. This will give an indication of what words tend to be used together. It might be useful for analyzing the vocabulary of a collection.

Classification and Prediction

Even though classification is a form of prediction, it deserves a separate category within text mining because it is such a common task. Classification involves sorting the documents into predefined categories and requires a preclassified training set in order to accomplish it. Nodes in Enterprise Miner that assist in classifying documents include Neural Network, Memory-Based Reasoning, and Tree nodes. Applications of classification include automatic e-mail routing, pushing documents to a user based on his or her profile, filtering spam, and matching resumes with open positions.

One last but diverse goal of text mining is prediction. The prediction of a target variable from the contents of text also requires a training data set. The application of prediction with textual data varies a great deal. Here are some examples:

- predicting the change in a stock price from contents of news announcements about companies
- predicting the cost of a service call based on the textual description of the problem
- predicting customer satisfaction from customer comments
- identifying authorship from a predetermined set of candidate authors.

In Enterprise Miner, all the modeling nodes support the function of predicting the target variable. See the respective node documentation for more information about each node.

CHAPTER 7

Examples

Example 1: Classifying News Articles **39**
Creating the Process Flow Diagram **39**
Viewing Results in Model Manager **40**
Viewing Output Variables in the Data Set Attributes Node **42**
Using SAS Code Node to Generate a Precision and Recall ROC Chart **43**
Precision and Recall ROC Chart **45**
PRCRCROC and PREREC Macros **46**
Example 2: Scoring New Documents **46**

Example 1: Classifying News Articles

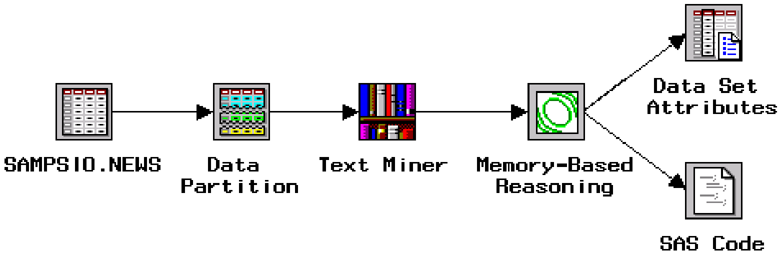
The SAMPSIO.NEWS data set consists of 600 brief news articles. Most of the news articles fall into one of the possible three categories: computer graphics, hockey, and medical issues.

Each article constitutes an observation. Therefore, SAMPSIO.NEWS data set contains 600 observations and the following variables:

- □ TEXT — a nominal variable that contains the text of the news article.
- □ GRAPHICS — a binary variable that indicates whether the document belongs to the computer graphics category (1-yes, 0-no).
- □ HOCKEY — a binary variable that indicates whether the document belongs to the hockey category (1-yes, 0-no).
- □ MEDICAL — a binary variable that indicates whether the document is related to medical issues (1-yes, 0-no).

Creating the Process Flow Diagram

Follow these steps to create the Process Flow Diagram:



Input Data Source Node

1 Add an Input Data Source node to the diagram workspace.

2 Open the node and set SAMPSIO.NEWS as the source data set.

3 Select the Variables tab and assign a role of **`input`** to the variable TEXT.

4 Set the model role of the variable HOCKEY to **`target`** and those of GRAPHICS and MEDICAL to **`rejected`**.

5 Right-click HOCKEY and select **`Edit target profile`**. Then, in the Assessment Information tab, select **`Default profit`** and set it to **`use`**. Close the target profiler and save the changes.

6 Close the *Input Data Source* node and save the changes.

Data Partition Node

1 Add a Data Partition node to the diagram workspace and connect it to the Input Data Source node.

2 Open the node.

3 In the Partition tab, change the percentage values for the training, validation, and test data sets to 60%, 20%, and 20%, respectively.

4 Close the Data Partition node and save the changes.

Text Miner Node

1 Add a Text Miner node to the diagram workspace and connect it to the Data Partition node.

2 Open the Text Miner Settings window.

3 In the Parse tab, ensure that the value of **`language`** is set to English and that **`Same word as different part of speech`**, **`Stemmed words as root form`**, and **`Noun group`** are selected.

4 In the Transform tab, change the value of **`Maximum Dimensions`** to 50.

5 Close the Text Miner node and save the changes. Run the Text Miner node.

Memory-Based Reasoning Node

1 Add a Memory-Based Reasoning (MBR) node to the diagram workspace and connect it to the Text Miner node.

2 Open the Memory-Based Reasoning node.

3 In the Output tab, select the **`Process or Score:  Training, Validation, and Test`** check box.

4 Close the node and save the model as an **`MBR Model`**.

5 Run the Memory-Based Reasoning node.

Viewing Results in Model Manager

1 After the Memory–Based Reasoning node has run successfully, right-click the node and select **`Model Manager`**.

2 In the Model Manager, click the Draw Diagnostic Chart tool icon 📊 to create a diagnostic chart for **`MBR Model`**.

By default, Model Manager uses the validation data set to create assessment charts. In this example flow, the validation data set accounts for 20% of the input data and has 120 observations.

The following display shows the diagnostic chart of **`MBR Model`**:

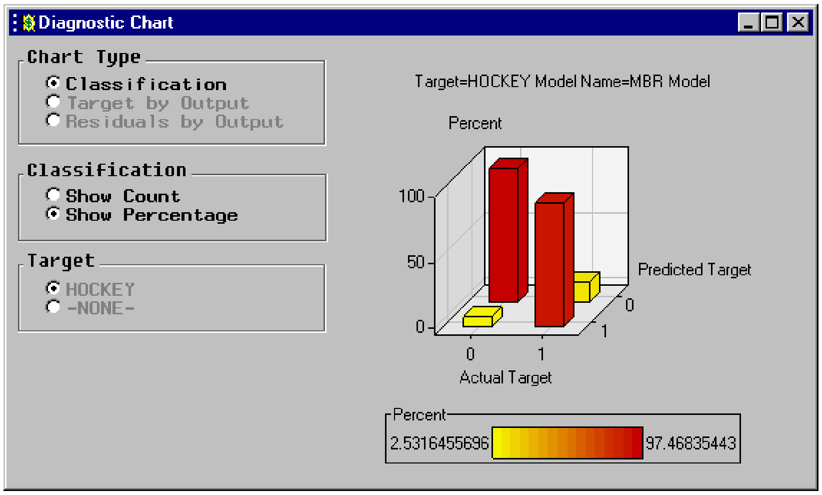


Chart Type
● Classification
○ Target by Output
○ Residuals by Output

Classification
○ Show Count
● Show Percentage

Target
● HOCKEY
○ -NONE-

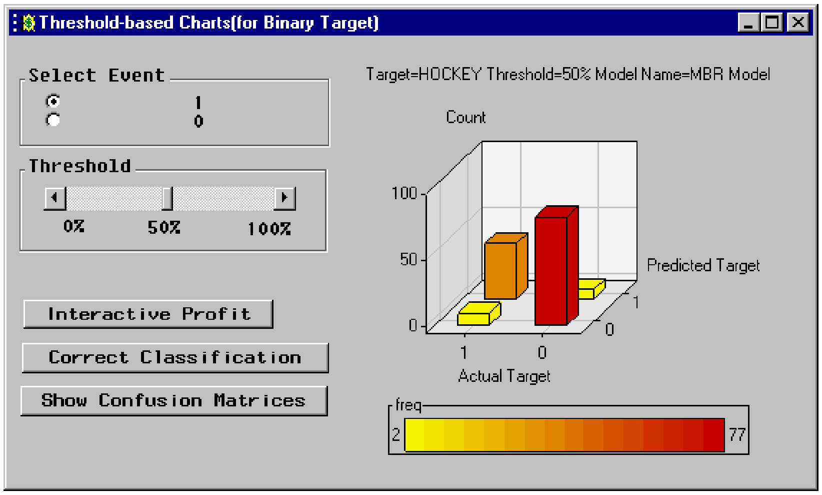
The diagnostic chart is a classification chart that displays the agreement between the predicted and actual target values. This example displays the percentages. 90.24% of the articles (37 out of 41) that actually belong to the hockey category are correctly predicted. Of the articles that do not belong to the hockey category, 97.47% of them (77 out of 79) are correctly classified.

3 Close the Diagnostic Chart window.

4 In the Model Manager, select from the main menu

Tools ▶ Threshold-Based Chart

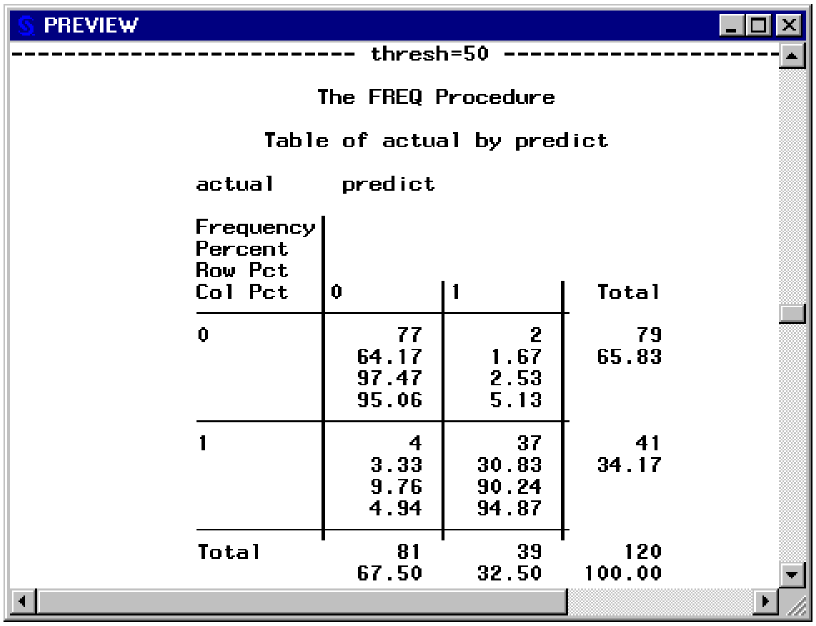
The threshold-based chart displays the agreement between the predicted and actual target values at various threshold levels between 0% and 100%. Use the threshold slider or the arrows to change the threshold level. The default threshold level is 50%. Therefore, the initial threshold-based chart displays the same information as the previous diagnostic chart.



Select Event
● 1
○ 0

Threshold
0% 50% 100%

Click the Show Confusion Matrices button to display confusion matrices for threshold values that range from 0% to 100%, arranged in 5% increments. The following display shows the confusion matrix with a threshold of 50% for the scored validation data set:



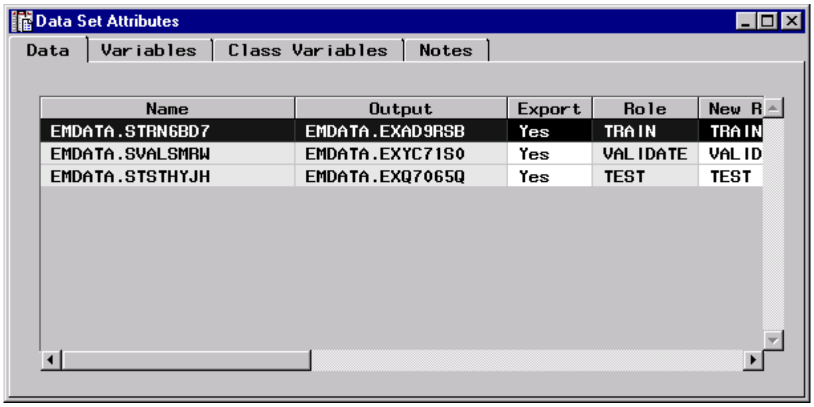
```
------------------------- thresh=50 ---------------------

                  The FREQ Procedure

              Table of actual by predict

        actual     predict

        Frequency|
        Percent  |
        Row Pct  |
        Col Pct  |0        |1        |  Total
        ---------+---------+---------+
        0        |     77  |      2  |     79
                 |  64.17  |   1.67  |  65.83
                 |  97.47  |   2.53  |
                 |  95.06  |   5.13  |
        ---------+---------+---------+
        1        |      4  |     37  |     41
                 |   3.33  |  30.83  |  34.17
                 |   9.76  |  90.24  |
                 |   4.94  |  94.87  |
        ---------+---------+---------+
        Total          81        39      120
                    67.50     32.50   100.00
```

Interpretation of this confusion matrix is similar to that of the diagnostic chart:

- 77 articles (out of 120) or 64.17% of the articles in the validation data set are correctly predicted as 0. That is, they are correctly predicted not to be in the hockey category.
- 37 articles (out of 120) or 30.83% of articles in the validation data set are correctly predicted as 1. That is, they are correctly predicted to be in the hockey category.
- Of all the articles that are classified into the hockey category (predict=1), 37 (or 94.87%) of them are correctly predicted.
- Of all the articles that actually belong to the hockey category, 4 (or 4.94%) of them are misclassified.
- The overall correct classification rate is (37+77)/120 = 95%.

Viewing Output Variables in the Data Set Attributes Node

1. Add a Data Set Attributes node to the diagram workspace and connect it to the Memory-Based Reasoning node.
2. Open the Data Set Attributes node.
3. The Data tab shows the data sets that are output from the Memory-Based Reasoning node:



Name	Output	Export	Role	New R
EMDATA.STRN6BD7	EMDATA.EXAD9RSB	Yes	TRAIN	TRAIN
EMDATA.SVALSMRW	EMDATA.EXYC71S0	Yes	VALIDATE	VALID
EMDATA.STSTHYJH	EMDATA.EXQ7065Q	Yes	TEST	TEST

Click a row. For example, click the scored training data set to select the data set.

4 Select the Variables tab to view variables in the scored training data set. The scored training data set contains the variables in the original training data set, the SVD dimension variables (COL1–COL50), and other variables as shown in the following list.

- □ P_HOCKEY0 — the posterior probability that an observation is predicted into 0.
- □ P_HOCKEY1 — the posterior probability that an observation is predicted into 1.
- □ D_HOCKEY_ — the label of the decision that is chosen by the model.
- □ EP_HOCKEY_ — the expected profit for the decision that is chosen by the model.
- □ BP_HOCKEY_ — the best possible profit of the decision.
- □ CP_HOCKEY_ — the profit that is computed from the target value
- □ I_HOCKEY — the target level that an observation is classified into.
- □ U_HOCKEY — the unnormalized target level that an observation is classified into.
- □ _WARN_ — a character variable that indicates problems in computing predicted values or making decisions. A blank value indicates that there is no problem.
- □ _DOCUMENT_ — the document ID.
- □ _SVDLEN_ — the length of the k-dimensional vector of SVD dimensions before the dimensions are normalized.
- □ _SVD_1 — _SVD_50 — the SVD dimensions.

Using SAS Code Node to Generate a Precision and Recall ROC Chart

1 Add a SAS Code node to the diagram workspace and connect it to the second Memory-Based Reasoning node, which uses HOCKEY as the target.

2 Open the SAS Code node.

3 Select the Program tab and enter the following SAS code:

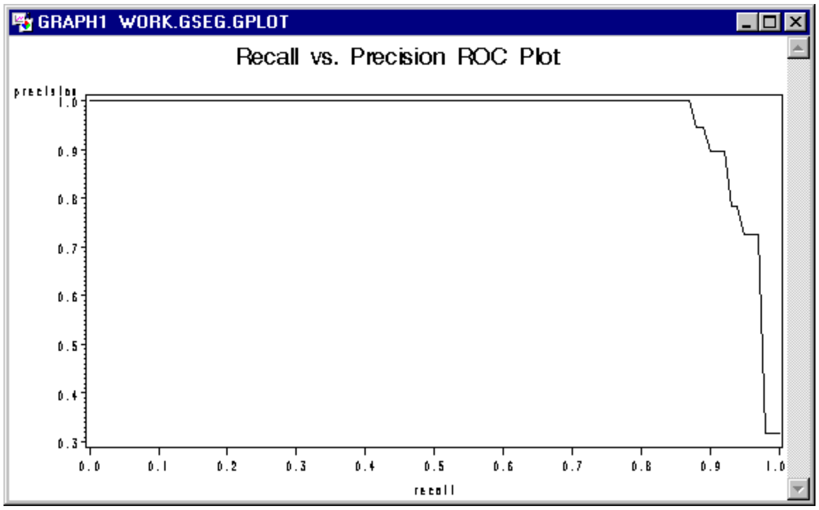
```
%prcrcroc(hockey, P_hockey1, &_test);
%prerec(hockey, D_hockey_, &_test);
run;
```

The macro PRCRCROC creates a precision and recall ROC plot. The macro PREREC prints a confusion matrix of the scored test data set. For more information about these macros, see “PRCRCROC and PREREC Macros” on page 46.

4 Close the SAS Code node and save the changes.

5 Run the SAS Code node and click Yes to view the results.

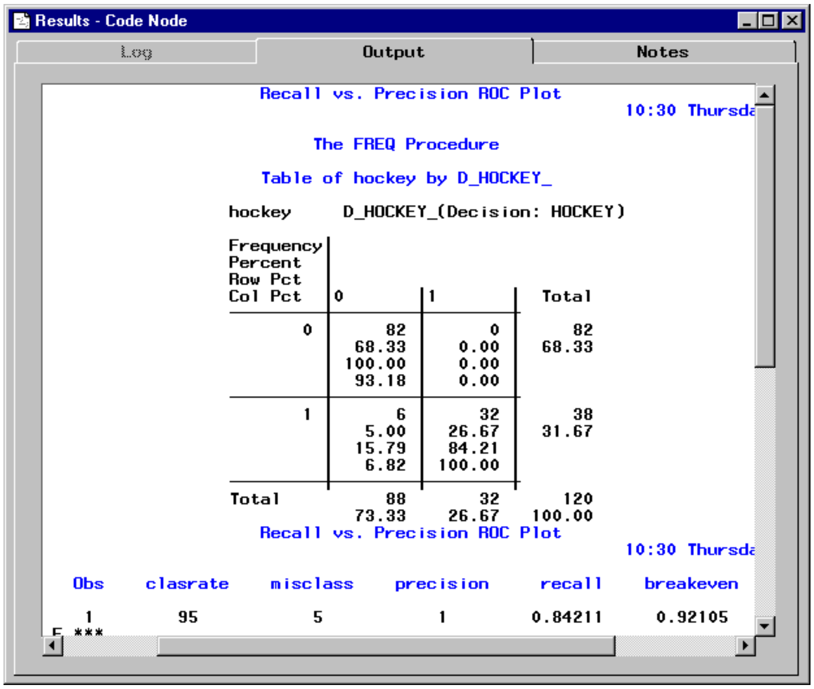
6 The following display shows the precision and recall ROC chart:



This chart that pushes upward and to the right illustrates a good prediction model.

For more information, see “Precision and Recall ROC Chart” on page 45.

7 The Output tab of the SAS Code Results Browser displays the confusion matrix (with a threshold value of 50%) of the scored training data set. The scored test data set has 120 observations, which is 20% of the original input data set (600). The following display shows the resulting confusion matrix.



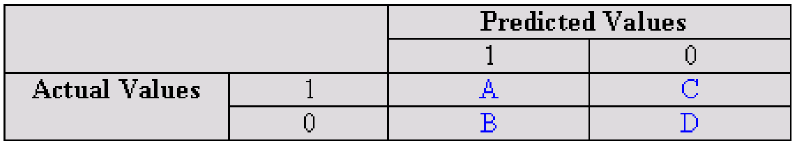
```
                  Recall vs. Precision ROC Plot
                                                      10:30 Thursda

                       The FREQ Procedure

                  Table of hockey by D_HOCKEY_

             hockey     D_HOCKEY_(Decision: HOCKEY)

             Frequency|
             Percent  |
             Row Pct  |
             Col Pct  |0      |1      |  Total
             ---------+-------+-------+
                    0 |    82 |     0 |     82
                      | 68.33 |  0.00 |  68.33
                      |100.00 |  0.00 |
                      | 93.18 |  0.00 |
             ---------+-------+-------+
                    1 |     6 |    32 |     38
                      |  5.00 | 26.67 |  31.67
                      | 15.79 | 84.21 |
                      |  6.82 |100.00 |
             ---------+-------+-------+
             Total          88      32      120
                         73.33   26.67   100.00
                  Recall vs. Precision ROC Plot
                                                      10:30 Thursda

   Obs    clasrate    misclass    precision    recall    breakeven

    1        95           5           1        0.84211    0.92105
F ***
```

Interpretation of the matrix shows that

- 82 articles (out of 120) or 68.33% of the articles are correctly predicted as 0.
- 32 or 100% out of the articles (32) that are classified into the hockey category (D_hockey_= 1) are correctly predicted.
- Of all the articles (38) that actually belong to the hockey category, 6 (or 15.79%) are misclassified.
- The overall correct classification rate is 95%.
- Precision and recall are 1 and 0.84, respectively. The break-even point is 0.92, which is the average of these values.

Precision and Recall ROC Chart

Precision and recall are measures that describe the effectiveness of a binary text classifier to predict documents that are relevant to the category. A relevant document is one that actually belongs to the category. A classifier has a high precision if it assigns a low percentage of nonrelevant documents to the category. *Recall* indicates how well the classifier can find relevant documents and assign them to the correct category. Precision and recall can be calculated from a two-way contingency table:



		Predicted Values	
		1	0
Actual Values	1	A	C
	0	B	D

Suppose that the target value 1 is of interest, that *A* is the number of documents that are predicted into category 1 and actually belong to that category, that *A*+*C* is the number of documents that actually belong to category 1, and that *A*+*B* is the number of documents that are predicted into category 1. Then

$$\text{Precision} = \frac{A}{A+B}$$

$$\text{Recall} = \frac{A}{A+C}$$

Obtaining both high precision and high recall are generally mutually conflicting goals. To obtain high precision, the classifier assigns to the category only the documents that are definitely in the category. High precision is achieved at the expense of missing some documents that might also belong to the category, and it therefore lowers the recall.

The precision and recall ROC chart enables you to make a decision about a cutoff probability to categorize the documents. Charts that push upward and to the right represent good precision and recall, which means a good prediction model. The precision and recall ROC chart emphasizes the trade-off between precision and recall. The precision and recall ROC chart is relevant to the sensitivity and specificity ROC chart in the Assessment node, but it is not exactly the same. For more information about ROC charts, see the Assessment node documentation.

PRCRCROC and PREREC Macros

Two macros, PRCRCROC and PREREC, are used in this example to explore the results from the Memory-Based Reasoning node.

The macro PRCRCROC computes and plots a precision and recall curve for the scored data set. Here is an example of PRCRCROC:

```
%prcrcroc(hockey, P_hockey1, &_test);
```

In the example, hockey is the target variable, P_hockey1 is the posterior probability for an observation that the predicted value of hockey is 1, and &_test is the macro variable name of the scored test data set.

The macro PREREC is used to generate a tabular view of classification results. The following code shows an example:

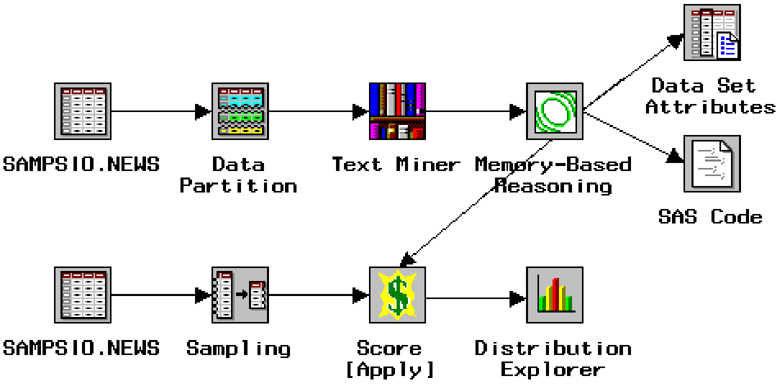
```
%prerec(hockey, D_hockey_, &_test)
```

In the example, hockey is the target variable, I_hockey is the label of the predicted category of an observation, and &_test is the scored test data set.

Example 2: Scoring New Documents

When the model for the document collection is generated, you can use it to predict the categories of new documents. Scoring can be done either at the same time as training or after the training is complete.

Suppose you have created a model as described in Example 1, and you want to score new documents with the model. In this case, you must use the Score node in your process flow diagram as follows.



In this example, a sample from the SAMPSIO.NEWS data set is used as the score data set. Follow these steps to score new documents.

Input Data Source Node

1 Add another Input Data Source node to the diagram workspace.
2 Open the node and set SAMPSIO.NEWS as the source data set.
3 In the Data tab, change the data role to **Score**.
4 Select the Variables tab and assign a role of **input** to the variable TEXT.
5 Set the model role of the variables GRAPHICS, HOCKEY, and MEDICAL to **rejected**.

6 Close the *Input Data Source* node and save the changes.

Sampling Node

1 Add a Sampling node to the diagram workspace and connect it to the second Input Data Source node. By default, the Sampling node generates a simple random sample from the input data set and the sample size is 10% of the size of the input data.

2 Run the Sampling node with default settings. This action creates a sample of size 60.

Score Node

1 Add a Score node to the diagram workspace and connect it to the Memory–Based Reasoning and Sampling nodes.

2 Open the Score node. In the Settings tab, select the **`Apply training data score code to score data set`** radio button.

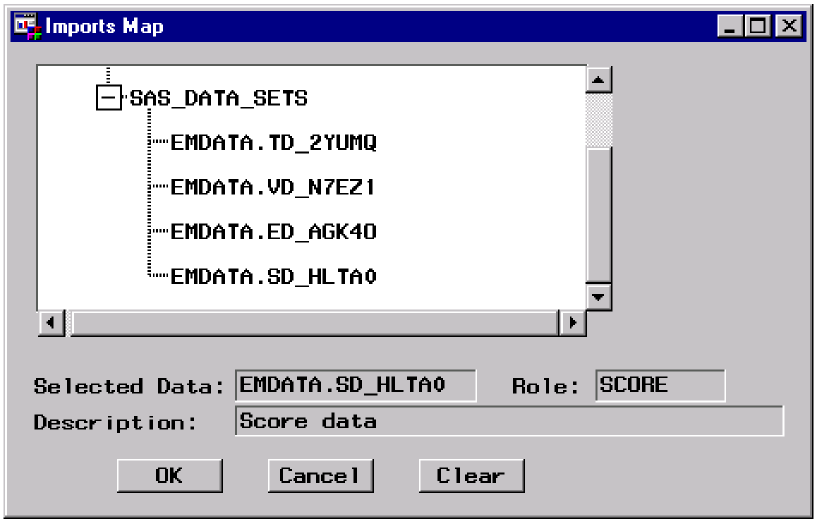
3 Run the Score node.

Distribution Explorer Node

1 Add a Distribution Explorer node to the diagram workspace and connect it to the Score node.

2 Open the Distribution Explorer node. In the Data tab, click Select to open the Import Map window that enables you to select a data set.

3 In the Imports Map window, select the score data set (EMDATA.SD_xxxx). Click OK .



Selected Data: EMDATA.SD_HLTA0 Role: SCORE

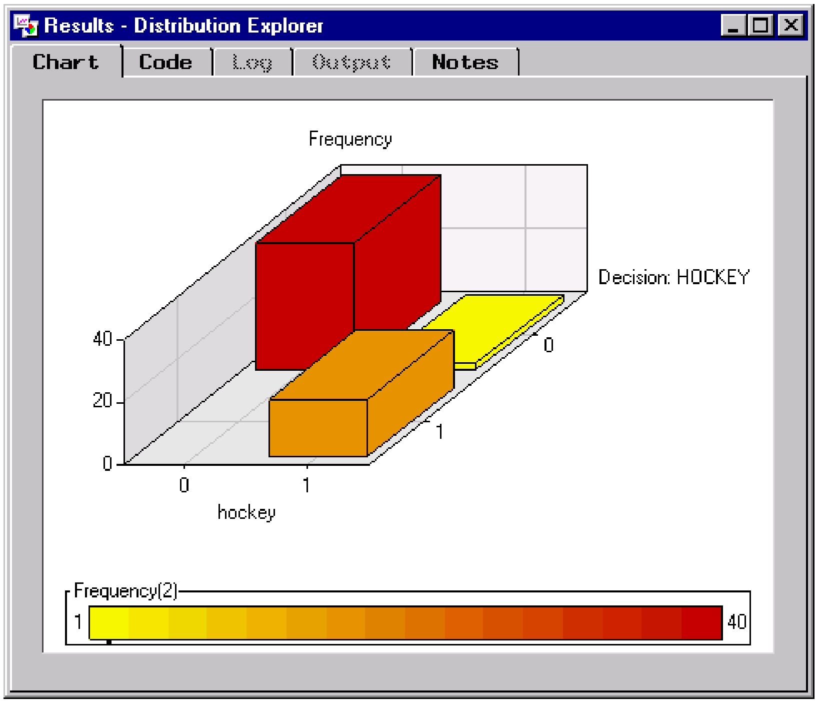
Description: Score data

4 In the Variables data, right-click the variable HOCKEY in the Axis column and select **`Set Axis`** and then **`X`**. Similarly, set the variable D_HOCKEY_, the predicted HOCKEY category for each document, as the **`Y`** axis. Run the node.

5 Open the Distribution Explorer Results Browser. The Chart tab displays a plot of the variables HOCKEY and D_HOCKEY_. Select

View ▶ Axes Statistics ▶ Response Axis ▶ Frequency

from the main menu. The following display shows the plot. 2, or 3.33%, of the 60 documents in the score data set are misclassified.



Alternatively, scoring can be done at the same time as training the model. To do this, you need to pass the score data set to the Text Miner node before running the Text Miner node or the Memory-Based Reasoning node in Example 1. Also, ensure that the **`Process or Score: Score`** check box in the Output tab in the Memory-Based Reasoning node is selected. In this case, running the modeling node will also score the new documents.

APPENDIX 1

References

References **49**

References

- Berry, M. W. and M. Browne. 1999. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Philadelphia: Society for Industrial and Applied Mathematics.
- Bradley, P.S., U.M. Fayyad, and C.A. Reina. 1998. Microsoft Research Technical Report MSR-TR-98–35. Microsoft Corporation. *Scaling EM (Expectation-Maximization) Clustering to Large Database*. Redmond, WA: Microsoft Corporation.
- Deerwester, et al. 1990. "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science*. 41(6): 391–407.
- Johnson, R.A. and D.W. Wichern. 1992. *Applied Multivariate Statistical Analysis*. Englewood Cliffs, NJ: Prentice-Hall.
- Krishnaiah, P.R. and L.N. Kanal. 1982. *Classification, Pattern Recognition, and Reduction of Dimensionality*. New York: North-Holland Publishing Company.
- McLachlan, G.J. 1997. *The EM Algorithm and Extensions*. New York: John Wiley & Sons.
- Trefethen, L.N. and D. Bau. 1997. *Numerical Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics.
- Watkins, D.S. 1991. *Fundamentals of Matrix Computations*. New York: John Wiley & Sons.
- Yang, Y. and J.O. Pedersen. 1997. "A Comparative Study on Feature Selection in Text Categorization." *Proceedings of the Fourteenth International Conference on Machine Learning* (ICML'97).

APPENDIX 2

Recommended Reading

Recommended Reading 51

Recommended Reading

Here is the recommended reading list for this title:

□ *Data Mining Using SAS Enterprise Miner: A Case Study Approach*
□ *Getting Started with SAS Enterprise Miner*

For a complete list of SAS publications, see the current *SAS Publishing Catalog*. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: (800) 727-3228*
Fax: (919) 677-8166
E-mail: **sasbook@sas.com**
Web address: **support.sas.com/pubs**
* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.

Index

A

ASCII text
  converting text files to data sets 15
automatic stemming 24

B

binary weight 33

C

canonical forms
  entities of 25
Chi-Squared weighting 33
classification 37, 38
  example 39
classifying news articles 39
CLUSTER procedure 37
Cluster Settings window 10
clustering 37
  expectation-maximization clustering 37
  hierarchical clustering 37
clustering analysis 10
Clustering node 38
Clusters table 10
customizing equivalent terms 26
  entities 29
  noun groups 29
  part-of-speech category 27
  synonyms 27, 29

D

Data Set Attributes node
  viewing output variables in 42
data sets
  converting text files to 16
  creating with %tmfilter 15
descriptive mining 1
dimension reduction 34
  roll-up terms 34
  singular value decomposition 34
document formats 15
documents
  frequency of a term 31
  quantitative representation for 31
  scoring new documents 46
Documents table 8

E

English
  default stop list 23
Enterprise Miner 2
entities
  customizing equivalent terms 29
  ISO standards for 25
  of canonical forms 25
  parsing 21
entity category 22, 23
Entropy weight 33
equivalent terms 24
  automatic stemming 24
  customizing 26
  entities of canonical forms 25
  synonyms 25
examples 5, 39
expectation-maximization clustering 37
exploring data 1
external files
  as text source 15
  document formats for 15

F

foreign languages
  parsing 22
French
  default stop list 23
frequency of terms 31
frequency weights 32

G

German
  default stop list 23
global frequency times inverse document frequency 33

H

hierarchical clustering 37

I

Information Gain weighting 33
information theory 33
Input Data Source node 7
inverse document frequency 33

L

languages
  parsing 22
log weight 33

M

macros
  PRCRCROC 46
  PREREC 46
  %tmfilter 2, 15, 16
Model Manager 40
multiword terms and phrases 21
Mutual Information weighting 33

N

news articles, classifying 39
normal weight 33
noun groups
  customizing equivalent terms 29

O

output variables
  viewing in Data Set Attributes node 42

P

parsed terms
  examples of 20

parsing text
  See text parsing
part-of-speech category
  customizing equivalent terms 27
  parsing 22
PRCRCROC macro 46
precision and recall ROC charts 43, 45
prediction 37, 38
predictive mining 1
predictive modeling 1
PREREC macro 46
process flow diagrams
  building 5
  creating 7, 39
  examples 5, 39
projects
  creating 6

Q

quantitative representation for documents 31

R

Results Browser 8
roll-up terms 34

S

SAS Code node
  precision and recall ROC charts 43
SAS Text Miner 2
  examples 5, 39
scoring new documents 46
singular value decomposition 34
SOM/Kohonen node 38
sources of text 15
start lists 23
stemming 24
stop lists 23
SUGI papers example 5
supervised learning 37
synonym list 25
  customizing equivalent terms 26
synonyms 25
  customizing equivalent terms 27, 29

T

taxonomy 37
term-by-document frequency matrix 31
  weighting entries 32
term importance weightings 33
term weights 32, 33
terms
  frequency in a document 31
Terms table 8
text files
  converting to data sets 16
Text Miner node 2, 7
Text Miner Results Browser 8
text mining 1
  process of 3
  tips for 3
text parsing 19
  entities 21
  entity category 22, 23
  equivalent terms 24
  examples of parsed terms 20
  foreign languages 22
  multiword terms and phrases 21
  part-of-speech category 22
  quantitative representation for documents 31
  start lists 23
  stop lists 23
text sources 15
tfidf 33
%tmfilter macro 2
  converting text files to data sets 16
  creating data sets 15
transformation 34

U

unsupervised learning 37

W

Ward's minimum-variance method 37
weighting 32

Your Turn

If you have comments or suggestions about *Getting Started with SAS 9.1 Text Miner*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
email: `yourturn@sas.com`

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: `suggest@sas.com`