SAS® Text Miner 12.1 Reference Help

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Text Miner 12.1 Reference Help*. Cary, NC: SAS Institute Inc.

SAS® Text Miner 12.1 Reference Help

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

1st electronic book, August 2012

Contents

Chapter 1
Introduction to SAS Text Miner and Text Mining

What's New in SAS Text Miner 12.1

Overview of What’s New

SAS Text Miner 12.1 includes the following new features and enhancements:

- “New Text Rule Builder Node” on page 2
- “Enhancements to Text Mining Nodes” on page 2
- “New Sample Data Set” on page 2

For more information, see `http://support.sas.com/software/products/txtminer`.

New Text Rule Builder Node

The new **Text Rule Builder** node enables you to do predictive modeling directly from the term-by-document matrix, thereby allowing user-assisted or "active" learning. You can use the **Text Rule Builder** node to create rules that can be exported to SAS Content Categorization Studio. For more information about the **Text Rule Builder** node, see "Overview of the Text Rule Builder Node" on page 99.

Enhancements to Text Mining Nodes

Improvements to previously existing text mining nodes include enhancements to the **Text Filter** node and viewer, the **Text Topic** node and viewer, and the **Text Cluster** node.

New Sample Data Set

The AFINN_SENTIMENT data set in the SAMPSIO library is new in this release. It is adapted from the AFINN sentiment publically available English sentiment lexicon. It contains two topics, "Positive Tone" and "Negative Tone," that can be used as User Topics in the **Text Topic** node. The AFINN_SENTIMENT data set contains information from the **AFINN sentiment database**, which is made available under the **Open Database License**.

Replacing the Original Text Miner Node

The functionality that was available in the original **Text Miner** node has been moved to other nodes available with SAS Text Miner. This restructuring of functionality conforms more to the overall philosophy of SAS Enterprise Miner components and improves performance since you can make changes in nodes that follow the **Text Parsing** node without having to reparse the collection. The following table might be helpful to you in replacing the functionality that you are using in the original **Text Miner** node with the new nodes.

Controls and Functionality in the Original Text Miner Node	Replacement in New SAS Text Miner nodes
Parsing	**Text Parsing** node
Term weightings	**Text Filter** node
Concept Linking Diagram	**Text Filter** node. The concept linking diagram is available in the Interactive Filter Viewer.
Creating and removing synonyms interactively	**Text Filter** node
Dynamically keeping and dropping terms	**Text Filter** node
Subsetting data for reclustering	**Text Filter** node

Controls and Functionality in the Original Text Miner Node	Replacement in New SAS Text Miner nodes
Clustering (Expectation Minimization and Hierarchical)	**Text Cluster** node
Generation of SVD values	**Text Cluster** node. Use this node output as input to your predictive models, for example.
Roll-up Terms	**Text Topic** node. Set the number of single term topics to the number of Roll-up Terms that you desire.

Accessibility Features of SAS Text Miner 12.1

Overview of Accessibility Features

SAS Text Miner 12.1 includes accessibility and compatibility features that improve the usability of the product for users with disabilities, with exceptions noted below. These features are related to accessibility standards for electronic information technology that were adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended. SAS Text Miner 12.1 conforms to accessibility standards for the Windows platform.

For specific information about Windows accessibility features, refer to your operating system's help. If you have questions or concerns about the accessibility of SAS products, send e-mail to `accessibility@sas.com`.

For information about the accessibility features of SAS Enterprise Miner 12.1, see the Accessibility topic in the SAS Enterprise Miner 12.1 Help.

- "Additional Keyboard Controls for SAS Text Miner" on page 3
- "Exceptions to Accessibility Standards" on page 4

Additional Keyboard Controls for SAS Text Miner

In addition to standard keyboard controls, SAS Text Miner supports the following additional keyboard controls.

Table 1.1 *Keyboard Controls for Tables*

Keyboard Shortcut	Action
Shift + Page Up	selects all rows in the table from the first row that is visible in the scroll pane to the selected row
Shift + Page Down	selects all rows in the table from the selected row to the last row that is visible in the scroll pane

Keyboard Shortcut	Action
Ctrl + Shift + End	selects all rows in the table from the selected row to the last row
Ctrl + Shift + Home	selects all rows in the table from the first row to the selected row

Table 1.2 *Other Keyboard Controls*

Keyboard Shortcut	Action
F6	selects the Properties panel
Tab	navigates the Properties panel
F2+Spacebar	simulates the click action for ellipsis in the Properties panel
Ctrl + Shift + n	select a node in a process flow diagram
Ctrl + Shift + c	connect selected nodes in a process flow diagram

Exceptions to Accessibility Standards

Exceptions to the accessibility standards described in Section 508 of the U.S. Rehabilitation Act of 1973 include the following:

- On-screen indication of the current focus is not well-defined in some dialog boxes, in some menus, and in tables.
- High-contrast color schemes are not universally inherited.
- Many controls are not read by JAWS, and the accessible properties of many controls are not surfaced to the Java Accessibility API.

About SAS Text Miner

SAS Text Miner provides tools that enable you to extract information from a collection of text documents and uncover the themes and concepts that are revealed therein. In addition, because you can embed SAS Text Miner nodes in a SAS Enterprise Miner process flow diagram, you can combine quantitative variables with unstructured text in the mining process and thereby incorporate text mining with other traditional data mining techniques.

These languages are supported in SAS Text Miner: Arabic, Chinese (simplified and traditional), Czech, Danish, Dutch, English, Finnish, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Romanian, Russian, Slovak, Spanish, Swedish, Thai, Turkish, and Vietnamese. Each language must be licensed individually.

Note: Collections of text in some unsupported languages can still be processed in SAS Text Miner by choosing a supported language that uses the same or similar text encoding as the unsupported language.

SAS Text Miner includes the following SAS Enterprise Miner nodes:

- **Text Import Node** — enables you to create data sets that contain links to documents obtained with file crawl, Web crawl, or Web search. For more information, see "Overview of the Text Import Node" on page 13.
- **Text Parsing Node** — enables you to parse a document collection in order to quantify information about the terms that are contained therein. For more information, see "Overview of the Text Parsing Node" on page 22.
- **Text Filter Node** — enables you to reduce the total number of parsed terms or documents that are analyzed. For more information, see "Overview of the Text Filter Node" on page 44.
- **Text Topic Node** — enables you explore the document collection by clustering documents and summarizing the collection into a set of "topics." For more information, see "Overview of the Text Topic Node" on page 83.
- **Text Cluster Node** — enables you to cluster documents from the term-document frequency matrix that is created by the **Text Parsing** node and possibly refined by the **Text Filter** node. For more information, see "Overview of the Text Cluster Node" on page 65.
- **Text Rule Builder Node** — enables you to generate rules that are useful in describing and predicting a target variable. For more information, see "Overview of the Text Rule Builder Node" on page 99.

In SAS Text Miner, the text mining process consists generally of the steps that are listed in the following table.

Step	Action	Description	Tools
1	File Preprocessing	Create a SAS data set from a document collection that is used as input for the **Text Parsing** node.	**Text Import** node, %TMFILTER macro, or SAS DATA step.

Step	Action	Description	Tools
2	Text Parsing	Decompose textual data, and generate a quantitative representation that is suitable for data mining purposes. Parsing might include: • stemming • automatic recognition of multi-word terms • normalization of various entities such as dates, currency, percent, and year • part-of-speech tagging • extraction of entities such as organization names, product names, and addresses • support for synonyms • language-specific analyses	**Text Parsing** node
3	Text Filtering	Transform the quantitative representation into a compact and informative format; reduce dimensions.	**Text Filter** node
4	Document Analysis	Cluster, classify, predict, or link concepts.	**Text Topic** node, **Text Cluster** node, **Text Rule Builder** node, and SAS Enterprise Miner predictive modeling nodes

TIP A number of data sets are provided that might be useful for learning how to use SAS Text Miner:

For more information about each action and sample data sets, see below.

- "File Preprocessing" on page 7
- "Text Parsing" on page 7
- "Text Filtering" on page 8

- “Document Analysis” on page 8
- “SAS Text Miner Sample Data Sets” on page 9

Note: SAS Text Miner 12.1 is not included with the base version of SAS Enterprise Miner 12.1. If your site has not licensed SAS Text Miner 12.1, then the SAS Text Miner nodes will not appear in your SAS Enterprise Miner 12.1 software.

File Preprocessing

The **Text Parsing** node requires an Input Data Source node to precede it in a process flow diagram. Input data for the **Text Parsing** node must be imported into a data source. Furthermore, because the **Text Parsing** node expects input data in a particular format, in most cases you will need to preprocess data before you can import it into a data source.

The **Text Import** node and the SAS %TMFILTER macro can be used to preprocess data.

The **Text Import** node can be used to extract text from many document formats or to retrieve text from Web sites by crawling the Web.

The SAS **%TMFILTER** macro can be used in file preprocessing to extract text from many document formats. You can use this macro to create a SAS data set that can be used to create a data source to use as input for the **Text Parsing** node. The SAS **%TMFILTER** macro does not extract data from individual XML fields. However, you can still accomplish this task. The XML LIBNAME engine and the XML mapper in Base SAS enable you to read an XML file into a SAS data set where the fields of the XML document are the data set variables. This data set can then be used in SAS Text Miner for further preprocessing.

Documents are represented internally in SAS Text Miner by a vector that contains the frequency of how many times each term occurs in each document. This approach is very effective for short, paragraph-sized documents but can cause a harmful loss of information with longer documents. Consider preprocessing long documents in order to isolate content that is really of use in the model that you intend to build. For example, if you are analyzing journal papers, you might find that analyzing only the abstracts gives the best results.

For more information about the **Text Import** node, the **%TMFILTER** macro, or the **Text Parsing** node, see the following:

- “Overview of the Text Import Node” on page 13
- “%TMFILTER Macro” on page 138
- “Overview of the Text Parsing Node” on page 22

Text Parsing

In SAS Text Miner, text parsing is done with the **Text Parsing** node. Advanced techniques enable you to break documents into terms such as words, phrases, multi-word terms, entities, punctuation marks, and terms that are in foreign languages.

- You can process multi-word groups (for example, “off the shelf” or “because of”) as single terms.

- You can identify each term's part of speech, based on its context.
- You can extract entities such as addresses, dates, phone numbers, and company names.
- You can choose to ignore all terms that are a particular part of speech.
- You can use a stop list to ignore a specific set of terms, such as a group of low-information words. Conversely, you can use a start list to restrict parsing to only a specific set of terms.
- You can return the root forms (called stems) of terms and treat all terms that have the same stem as equivalent. For example, “grinds”, “grinding”, and “ground” could all be viewed as the term “grind”.
- You can specify synonyms (such as “teach”, “instruct”, “educate”, “train”), and treat them as equivalent.

For more information about the **Text Parsing** node, see the following:

- “Overview of the Text Parsing Node” on page 22

Text Filtering

In SAS Text Miner, text filtering is done with the **Text Filter** node.

- You can explore a parsed document collection with the **Interactive Filter Viewer** of the **Text Filter** node. For more information about the **Interactive Filter Viewer**, see “Interactive Filter Viewer” on page 60.
- You can subset collections of documents based on the attributes of a document or the content of the document.
- You can interactively adjust the stop list and synonyms to focus on the aspects of the collection that are of interest to you.
- For more information about the **Text Filter** node, see “Overview of the Text Filter Node” on page 44.

Document Analysis

Exploration

SAS Text Miner offers visualization diagrams, topic creation, and clustering techniques that enable you to explore a parsed document collection. Applications include content discovery of large knowledge bases such as those that contain e-mail, customer comments, abstracts, or survey data; unsupervised learning of categories; and taxonomy creation.

- You can generate data-driven topics and supply topics that you have defined in the **Text Topic** node for use in scoring new data.
- You can perform hierarchical clustering in the **Text Cluster** node. The node uses a Ward's minimum-variance method to generate hierarchical clusters, and results are presented in a tree diagram.

- You can perform expectation-maximization (EM) clustering in the **Text Cluster** node. EM clustering identifies primary clusters, which are the densest regions of data points, and secondary clusters, which are less dense groups of data points not included in the primary clusters. This is a spatial clustering technique that allows flexibility in the size and shape of clusters.

- You can use other SAS Enterprise Miner nodes for clustering, such as the Clustering and SOM/Kohonen nodes. For more information about these nodes, see the SAS Enterprise Miner Help.

Prediction

You can use SAS Enterprise Miner modeling capabilities to predict target variables, with applications that include the following:

- automatic e-mail routing
- filtering spam
- matching resumes with open positions
- 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

The **Text Rule Builder** node can be used for prediction.

See the SAS Enterprise Miner Help for information about how to use modeling nodes for target variable prediction.

SAS Text Miner Sample Data Sets

Sample Data

The following sample data sets are provided in the SAMPSIO library for use with SAS Text Miner 12.1.

Table 1.3 *Sample Data Sets*

Data Set	Description	Used In
Abstract	document collection of abstracts of conference papers	Input Data node

Data Set	Description	Used In
AFINN_SENTIMENT	This data set is adapted from the AFINN sentiment publically available English sentiment lexicon. It contains two topics, “Positive Tone” and “Negative Tone,” that can be used as User Topics in the **Text Topic** node. The AFINN_SENTIMENT data set contains information from the `AFINN sentiment database`, which is made available under the `Open Database License`.	**Text Topic** node
News	document collection of brief news articles	Input Data node
Tm_abstract_topic	user-defined topics	**Text Topic** node

Default Data Sets

The following data sets are used in the SAS Text Miner 12.1 nodes as default inputs for node properties (for example, stop lists or multi-term lists). They are all in the SASHELP library.

Table 1.4 *Default Data Sets*

Data Set	Description	Used In
<language>_multi	(where <language> is: Eng, Frnch, Germ, Ital, Port, or Span) multi-term lists for various languages	**Text Parsing** node
<language>stop	(where <language> is: Eng, Frch, or Grmn) stop lists for various languages	**Text Parsing** node
Engsynms	synonym list for the English language	**Text Parsing** node

Chapter 2

Converting Diagrams and Session Encoding

Converting SAS Text Miner Diagrams from a Previous Version

Previous releases of SAS Text Miner included a **Text Miner** node. The **Text Miner** node is not available on the **Text Mining** tab in SAS Text Miner 12.1. However, you can import a diagram from an earlier release of SAS Text Miner that has a **Text Miner** node in a process flow diagram.

Note: If you import a **Text Miner** node, you will not be able to change its property values or open its Interactive window.

For more information about project conversion from earlier versions of SAS Enterprise Miner to SAS Enterprise Miner 12.1, see the SAS Enterprise Miner Help.

SAS Text Miner and SAS Session Encoding

For computing, text is stored as numbers that are mapped to letters and symbols for display. This mapping process is called “encoding.” Many different encodings exist for different languages, operating systems, and purposes. For example, WLATIN1 is an encoding for Western European languages, and BIG5 encodes characters from the Traditional Chinese language.

SAS Text Miner uses the same encoding settings as the current SAS session. Unless the current SAS session uses UTF-8 encoding, you might not be able to process text correctly if it is in an encoding different from the current SAS session. Furthermore, you cannot view such data sets in the SAS Enterprise Miner Explorer. Instead, start SAS with the encoding settings that correspond to the encoding of the data set that you are analyzing. Alternatively, for some compatible encodings, you can transcode a data set from its encoding to the SAS session encoding. Start SAS and using DATA step, simply copy your data set to a new data set. The new data set will be automatically transcoded to the current session encoding. Note that, using this method, it is possible that some characters cannot be converted between encodings.

Since UTF-8 is capable of representing all text, you can instead run a UTF-8 SAS session if you are working with a data set in another encoding. If your current SAS session uses UTF-8 encoding and the SAS Text Miner input data set contains the entire text of each document in the collection, then the data is transcoded (converted to) UTF-8 encoding. However, it is possible that some characters cannot be transcoded; SAS Text Miner ignores these characters.

For information about SAS session encoding, see *SAS National Language Support (NLS): Reference Guide*.

If your documents reside on the file system, you can also transcode them to the UTF-8 SAS session encoding. If you run the **%TMFILTER** macro or the **Text Import** node in a UTF-8 SAS session, all processed text will be transcoded to UTF-8. It is possible that some characters cannot be transcoded into UTF-8; SAS Text Miner ignores these characters. If a document collection is in multiple encodings or in an encoding different from the current SAS session, then it is recommended that you transcode the documents using this method. For more information about the **%TMFILTER** macro, see "%TMFILTER Macro" on page 138.

Note: It is not recommended to mix documents from various languages in a single SAS Text Miner analysis. Instead, use the language detection feature of the %TMFILTER macro, separate the documents by language, and analyze the subcollections separately.

Chapter 3

The Text Import Node

Overview of the Text Import Node

The **Text Import** node serves as a replacement for an Input Data node by enabling you to create data sets dynamically from files contained in a directory or from the Web. The **Text Import** node takes an import directory containing text files in potentially proprietary formats such as MS Word and PDF files as input. The tool traverses this directory and filters or extracts the text from the files, places a copy of the text in a plain text file, and a snippet (or possibly even all) of the text in a SAS data set. If a URL is specified, the node will crawl Web sites and retrieve files from the Web and to the import directory before doing this filtering process. The output of a **Text Import** node is a data set that can be imported into the **Text Parsing** node.

In addition to filtering the text, the **Text Import** node can also identify the language that the document is in and take care of transcoding documents to the session encoding. For more on encoding and transcoding, see "SAS Text Miner and SAS Session Encoding" on page 11.

The **Text Import** node relies on the SAS Document Conversion Server installed and running on a Windows machine. The machine must be accessible from the SAS Enterprise Miner server via the host name and port number that were specified at install time.

Note:

- If you run the **Text Import** node in a UTF-8 SAS session, then the node attempts to transcode all filtered text to UTF-8 encoding so that the result data set can be used in a UTF-8 SAS session. In all other SAS session encodings, the **Text Import** node does not transcode the data but instead assumes that the input data is in the same encoding as the SAS session. For more information, see "SAS Text Miner and SAS Session Encoding" on page 11.
- The **Text Import** node is not supported for use in group processing (Start Groups and End Groups nodes).

Text Import Node Input Data

The **Text Import** node does not require a predecessor node in a process flow diagram.

Text Import Node Properties

Contents

Text Import Node General Properties

These are the general properties that are available on the **Text Import** node:

- **Node ID** — displays the ID that is assigned to the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Import** node added to a diagram will have the Node ID `TextImport`, and the second Text Import node added will have the Node ID `TextImport2`.

- **Imported Data** — accesses a list of the data sets imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Exported Data** — accesses a list of the data sets exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Import Node Train Properties

General Train Properties

These are the training properties that are available on the **Text Import** node:

- **Import File Directory** — Specifies the path to the directory that contains files to be processed. Click the ellipsis for this property to select a directory accessible by the server for import.

- **Destination Directory** — Specifies the path to the directory that will contain plain text files after processing. Click the ellipsis for this property to specify a destination directory that is accessible by the server.

- **Language** — Specifies the possible choices that the language identifier might choose from when assigning a language to each document. Click the ellipsis for this property to open the Language dialog box to specify one or more languages. Only languages that are licensed can be used.

- **Extensions** — Restricts the **Text Import** node to filtering only files that satisfy the provided file type. All file types that the SAS Document Converter supports are filtered when the setting is not specified. See SAS Document Conversion for more information.

- **Text Size** — Specifies the number of characters to use in the TEXT variable of the output data set. This variable can serve as a snippet when the size is small, or you can set the value to as large as 32000, so that as much text as possible is placed in the data set.

- “Web Crawl Properties” on page 16

Note: A user can potentially view the contents of the Import File and Destination directories from the Interactive Filter Viewer of the **Text Filter** node. If, however, the directories are not accessible from the client, such as with a UNIX server and a Windows client, the documents will not be viewable from the client. In order to use this feature in this case, the files would need to be moved to an accessible directory and the path updated in the data set.

Web Crawl Properties

- **URL** — Specifies the URL of an initial Web page to crawl.
- **Depth** — Specifies the number of recursive levels of the URL to crawl. A depth of 1 means return all the files linked to from the initial page. A depth of 2 means return the files from a depth of 1 and also all the files that are linked to from that set, and so on. The number of files retrieved grows exponentially, so use caution when increasing the depth.
- **Domain** — Specifies whether to process documents outside the domain of the initial Web page.
- **User Name** — Specifies the user name when the URL input refers to a secured Web site and requires a user name and password.
- **Password** — Specifies the password when the URL input refers to a secured Web site and requires a user name and password.

Note: Web crawl properties are only available if SAS Text Miner uses a Windows server.

Text Import Node Status Properties

These are the status properties that are displayed on the **Text Import** node:

- **Create Time** — time that the node was created.
- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.
- **Last Error** — error message, if any, from the last run.
- **Last status** — last reported status of the node.
- **Run time** — time at which the node was last run.
- **Run duration** — length of time required to complete the last node run.
- **Grid Host** — grid host, if any, that was used for computation.
- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner extension node. The value of this property is always `No` for the **Text Import** node.

Text Import Node Results

Contents

Results Window for the Text Import Node

After the Text Import node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status window that opens immediately after a successful run of the **Text Import** node.
- Click the **Results** icon on the main Toolbar.
- Right-click the **Text Import** node, and select **Results**.

The Results window for the **Text Import** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are the same as other nodes. However, the **View** menu does not include the selections **Assessment** or **Model**. Instead, it includes **Documents**, which access submenus that list the graphical results.

Note: You can access the SAS log from the **SAS Results** submenu of the **View** menu. This log can be a useful debugging tool.

Text Import Node Graphical Results

The following are the graphical results in the **Text Import** node Results window:

- The **Omitted/Truncated Documents** pie chart shows which documents were omitted or truncated. Position the mouse pointer over a sector to see the status of a sector, such as Truncated, and the frequency in a tooltip.
- The **Created/Accessed/Modified Dates by Frequency** scatter plot shows the frequency by which documents were created, modified, or accessed. Position the mouse pointer over a point to see the date and frequency of the action in a tooltip.
- The **Document Languages** pie chart shows the languages represented among imported documents. Position the mouse pointer over a sector to see frequency of each language in a tooltip.
- The **Document Lengths by Frequency** bar chart shows document size and frequency. Position the mouse pointer over a bar for a tooltip containing this information.

- The **Document Types** pie chart shows the types of documents represented among imported documents by file extension, such as `.pdf`. Position the mouse pointer over a sector to see the frequency of each file extension in a tooltip.

Select a sector, point, or bar to highlight corresponding information in the other graphical results windows.

Text Import Node SAS Output Results

The SAS output from the **Text Import** node includes summary information about the input variables.

Text Import Node Output Data

For information about output data for the **Text Import** node, see "Output Data for SAS Text Miner Nodes" on page 127.

Using the Text Import Node

Contents

You can use the **Text Import** node to import documents from a directory or the Web. See the following for examples of how to use the **Text Import** node.

- "Import Documents from a Directory" on page 18
- "Import Documents from the Web" on page 19

Import Documents from a Directory

This example assumes that SAS Enterprise Miner is running, the SAS Document Conversion server is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

Perform the following steps to import documents from a directory:

1. Select the **Text Mining** tab, and drag a **Text Import** node into the diagram workspace.

2. Click the ellipsis button next to the **Import File Directory** property of the **Text Import** node.

   A Select Server Directory dialog box appears.

3. Navigate to a folder that contains documents that you want to create a data set from, select it, and then click **OK**.

   Note: To see the file types that you want to select, you might need to select **`All Files`** in the type drop-down menu.

4. Click the ellipsis button next to the **Language** property.

   The Language dialog box appears.

5. Select one or more licensed languages in which to require the language identifier to assign each document's language, and then click **OK**.

6. (Optional) Specify the file types to process for the **Extensions** property. For example, if you want to look at only documents with a .txt and a .pdf extension, specify **`.txt .pdf`** for the **Extensions** property, and click **Enter** on your keyboard.

   Note: If you do not specify file types to process, the **Text Import** node will process all file types in the specified import file directory.

7. Right-click the **Text Import** node, and select **Run**.

8. Click **Yes** in the Confirmation dialog box.

9. Click **Results** in the Run Status dialog box when the node has finished running.

10. Examine results from the documents that you imported.

    You can now use the **Text Import** node as an input data source for your text mining analysis.

11. Select the **Text Mining** tab, and drag a **Text Parsing** node into the diagram workspace.

12. Connect the **Text Import** node to the **Text Parsing** node.

13. Right-click the **Text Parsing** node, and select **Run**.

14. Click **Yes** in the Confirmation dialog box.

15. Click **Results** in the Run Status dialog box when the node has finished running.

Import Documents from the Web

This example assumes that SAS Enterprise Miner is running, the SAS Document Conversion server is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

Note: Web crawling is supported only on Windows operating systems.

Perform the following steps to import documents from the Web:

1. Select the **Text Mining** tab, and drag a **Text Import** node into the diagram workspace.

2. Enter the uniform resource locator (URL) of a Web page that you want to crawl in the URL property of the **Text Import** node. For example, try *www.sas.com*.

3. Type *1* as the number of levels to crawl in the Depth property.

4. Set the **Domain** property to **`Unrestricted`**.

Note: If you want to crawl a password-protected Web site, set the **Domain** property to **Restricted**, and provide a user name for the **User Name** property, and a password for the **Password** property.

5. Right-click the **Text Import** node and select **Run**.

6. Click **Yes** in the Confirmation dialog box.

7. Click **Results** in the Run Status dialog box when the node has finished running.

Chapter 4
The Text Parsing Node

Overview of the Text Parsing Node

The **Text Parsing** node enables you to parse a document collection in order to quantify information about the terms that are contained therein. You can use the **Text Parsing** node with volumes of textual data such as e-mail messages, news articles, Web pages, research papers, and surveys. See the following for more information about the **Text Parsing** node.

- "Text Parsing Node Input Data" on page 22
- "Text Parsing Node Properties" on page 23
- "Text Parsing Node Results" on page 27
- "Text Parsing Node Output Data" on page 29
- "Using the Text Parsing Node" on page 29

For related topics that you might find useful when using the **Text Parsing** node, see the following:

- "Start Lists and Stop Lists" on page 34
- "Term Stemming" on page 35
- "Term Roles and Attributes" on page 36
- "Synonym Lists" on page 38
- "Multi-Term Lists" on page 41

Note: The **Text Parsing** node is not supported for use in group processing (Start Groups and End Groups nodes).

Text Parsing Node Input Data

The **Text Parsing** node must be preceded by one or more Input Data Source nodes, where each data source contains a document collection to parse, or one or more **Text Import** nodes. At least one data source must have the role Train. Others can have roles of Train, Valid, Test, or Score.

Each observation from the input data source or **Text Import** node represents an individual document in the document collection. This data can have one of two structures. It can either contain the entire text of each document, or it can contain paths to plain text or HTML files that contain that text.

- If the data source contains the entire text of each document, then this text must be stored in a character variable that is assigned the role Text. Note that a SAS variable can hold only 32KB of text. If any document in the collection is larger than that limit, then you should not use this data source structure.

Note: There are sample data sets that you can use to create data sources with this structure. For more information, see "SAS Text Miner Sample Data Sets" on page 9.

- If the data source contains paths to files that contain the document text, then these paths must be stored in a character variable that is assigned the role Text Location. The paths must be relative to the SAS Text Miner server. This structure can be used either for collections that contain smaller documents or documents that exceed the 32KB SAS variable limit.

  TIP

  - To help identify which link represents which document, you can include an additional character variable that contains truncated document text. Give this variable the role Text.
  - If there is a variable in the data set that contains the location of the unfiltered documents and if you assign this variable the role Web Address, you will be able to access the original source of each document in the Interactive Results viewer.

The **Text Import** node creates the proper roles that are necessary for the **Text Parsing** node. The SAS **%TMFILTER** macro is an alternative to using the **Text Import** node to preprocess textual data. This macro creates a data source from the textual data that can be included with the Input Data Source node. For more information about these options to preprocess data for the **Text Parsing** node, see "File Preprocessing" on page 7.

A successful run of the **Text Parsing** node requires at least one variable with the role Text or Text Location. If you have more than one variable with either role (that has a use status of Yes), then the longest of these variables is used.

Text Parsing Node Properties

Contents

Text Parsing Node General Properties

These are the general properties that are available on the **Text Parsing** node:

- **Node ID** — displays the ID that is assigned to the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Parsing** node added to a diagram will have the Node ID `TextParsing`, and the second **Text Parsing** node added will have the Node ID `TextParsing2`.
- **Imported Data** — accesses a list of the data sets imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do any of the following:
  - browse the data set

  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Exported Data** — accesses a list of the data sets exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Parsing Node Train Properties

General Train Properties

These are the training properties that are available on the **Text Parsing** node:

- **Variables** — accesses a list of variables and associated properties in the data source. Click the ellipsis button to open the Variables window. For more information, see "Text Parsing Node Input Data" on page 22.
- "Parse Properties" on page 24
- "Detect Properties" on page 24
- "Ignore Properties" on page 25
- "Synonyms Properties" on page 25
- "Filter Properties" on page 26

Parse Properties

- **Parse Variable** — (value is populated after the node is run) displays the name of the variable in the input data source that was used for parsing. Depending on the structure of the data source, this variable contains either the entire text of each document in the document collection or it contains paths to plain text or HMTL files that contain that text.
- **Language** — accesses a window in which you can select the language to use when parsing. Click the ellipsis button to open the Languages window. Only supported languages that are licensed to you are available for selection. For a list of supported languages, see "About SAS Text Miner" on page 4.

Detect Properties

- **Different Parts of Speech** — specifies whether to identify the parts of speech of parsed terms. If the value of this property is Yes, then same terms with different parts of speech are treated as different terms. For more information, see "Parts of Speech in SAS Text Miner" on page 36.
- **Noun Groups** — specifies whether to identify noun groups. If stemming is turned on, then noun group elements are also stemmed. For more information, see "Noun Groups in SAS Text Miner" on page 36.

- **Multi-word Terms** — (for all languages except Chinese, Japanese, and Korean) specifies a SAS data set that contains multi-word terms. For more information, see "Multi-Term Lists" on page 41. Default data sets are provided for several languages. For more information, see "SAS Text Miner Sample Data Sets" on page 9. You can edit these data sets or create your own. Click the ellipsis button to open a window in which you can do the following:
  - import a multi-word term data set
  - (if a multi-word term data set is selected) add, delete, and edit terms in the multi-term list
- **Find Entities** — specifies whether to identify the entities contained in the documents. Entity detection relies on linguistic rules and lists that are provided for many entity types; these are known as standard entities. Custom entity types can be created by you by using SAS Concept Creation for SAS Text Miner software. For more information, see "Entities in SAS Text Miner" on page 37.
  - `None` identifies neither standard nor custom entities.
  - `Standard` identifies standard entities, but not custom entities.
  - `Custom` identifies custom entities, but not standard entities.
  - `All` identifies both standard and custom entities.
- **Custom Entities** — specifies the path (relative to the SAS Text Miner server) to a file that has been output from SAS Concept Creation for SAS Text Miner and contains compiled custom entities. Valid files have the extension `.li`. No custom entity should have the same name as a standard entity. For more information, see "Entities in SAS Text Miner" on page 37.

Ignore Properties

- **Ignore Parts of Speech** — accesses a window in which you can select one or more parts of speech. Terms assigned these parts of speech will be ignored when parsing. Click the ellipsis button to open the Ignore Parts of Speech dialog box. Use the SHIFT and CTRL keys to make multiple selections. Terms with the selected parts of speech are not parsed and do not appear in node results.
- **Ignore Types of Entities** — (if the value of **Find Entities** is `Standard` or `All`) accesses a dialog box in which you can select one or more standard entities to ignore when parsing. For more information, see "Entities in SAS Text Miner" on page 37. Click the ellipsis button to open the Ignore Types of Entities window. Use the SHIFT and CTRL keys to make multiple selections. Terms with the selected entity types are not parsed and do not appear in node results.
- **Ignore Types of Attributes** — accesses a window in which you can select one or more attributes to ignore when parsing. For more information, see "Attributes in SAS Text Miner " on page 38. Click the ellipsis button to open the Ignore Types of Attributes dialog box. Use the SHIFT and CTRL keys to make multiple selections. Terms with the selected attribute types are not parsed and do not appear in node results.

Synonyms Properties

- **Stem Terms** — specifies whether to treat different terms with the same root as equivalent. For more information see "Term Stemming" on page 35.

- **Synonyms** — specifies a SAS data set that contains synonyms to be treated as equivalent. For more information, see "Synonym Lists" on page 38. Default data sets are provided for several languages. For more information, see "SAS Text Miner Sample Data Sets" on page 9. You can edit these data sets or create your own. Click the ellipsis button to open a window in which you can do the following:
  - import a synonym data set
  - (if a synonym data set is selected) add, delete, and edit terms in the synonym list

Filter Properties

- **Start List** — specifies a SAS data set that contains the terms to parse. If you include a start list, then the terms that are not included in the start list appear in the results Term table with a Keep status of **`N`**. For more information, see "Start Lists and Stop Lists" on page 34. Click the ellipsis button to open a window in which you can do the following:
  - import a start list data set
  - (if a start list is selected) add, delete, and edit terms in the start list
- **Stop List** — specifies a SAS data set that contains terms to exclude from parsing. If you include a stop list, then the terms that are included in the stop list appear in the results Term table with a Keep status of **`N`**. For more information, see "Start Lists and Stop Lists" on page 34. Default data sets are provided for several languages. For more information, see "SAS Text Miner Sample Data Sets" on page 9. You can edit these data sets or create your own. Click the ellipsis button to open a window in which you can do the following:
  - import a stop list data set
  - (if a stop list is selected) add, delete, and edit terms in the stop list

Text Parsing Node Report Properties

This is the report property that is available on the **Text Parsing** node:

- **Number of Terms to Display** — indicates the maximum number of terms to be displayed in the Results viewer. Terms are first sorted by the number of documents in which they appear, and then the list is truncated to the maximum number. If the value of this property is **`All`**, then all terms are displayed.

Text Parsing Node Status Properties

These are the status properties that are displayed on the **Text Parsing** node:

- **Create Time** — time that the node was created.
- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.
- **Last Error** — error message, if any, from the last run.
- **Last Status** — last reported status of the node.
- **Last Run Time** — time at which the node was last run.
- **Run Duration** — length of time required to complete the last node run.

- **Grid Host** — grid host, if any, that was used for computation.
- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner extension node. The value of this property is always `No` for the **Text Parsing** node.

Text Parsing Node Results

Contents

Results Window for the Text Parsing Node

After the **Text Parsing** node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status window that opens immediately after a successful run of the **Text Parsing** node.
- Click the Results icon on the main toolbar.
- Right-click the **Text Parsing** node, and select **Results**.

The Results window for the **Text Parsing** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are the same as other nodes. However, the **View** menu for the **Text Parsing** node Results window does not include the selections **Assessment** or **Model**. Instead, it includes **Terms**, which accesses a submenu that lists the graphical and tabular results.

Note: You can access the SAS log that was generated by the node processing from the **SAS Results** submenu of the **View** menu. This log can be a useful debugging tool.

Text Parsing Node Graphical and Tabular Results

The following are the graphical results in the **Text Parsing** node Results window:

- The **Number of Documents by Frequency** scatter plot displays the number of documents in which a term appears versus the frequency of occurrence of that term in the entire document collection. Each data point represents a parsed term. If you position the mouse pointer over a plotted point, then a tooltip indicates the term name, the number of documents in which that term appears, and the number of times that term appears in the entire document collection.
- The **Role by Freq** bar chart displays the total frequency of occurrence of parsed terms in the document collection, broken down by term role. Each bar represents a role. If you position the mouse pointer over a bar, then a tooltip indicates the role name and the number of times a parsed term with that role appears in the entire document collection.

- The **Attribute by Frequency** bar chart displays the total frequency of occurrence of parsed terms in the document collection, broken down by attribute. For more information, see "Attributes in SAS Text Miner " on page 38. If you position the mouse pointer over a bar, then a tooltip indicates the attribute name and the number of times a term with that attribute appears in the entire document collection.
- The **ZIPF Plot** displays a scatter plot of the number of documents for each term where each is sorted and plotted by rank. If you position the mouse pointer over a bar, then a tooltip indicates the term name, rank, and number of documents.

The tabular result in the **Text Parsing** node Results window is the Terms table, which displays information about parsed top-level terms (in other words, terms that have no parents above them). Note that if there are more terms than the setting of the property Number of Terms to Display, only that number of most frequent terms will be included. All graphical results in the **Text Parsing** node Results window are linked to this table. Therefore, you can select an observation in the Terms table, and the associated data points are highlighted in the graphics. Or, you can select data points in the graphics, and the associated observations are highlighted in the Terms table.

Table 4.1 *Contents of the Terms Table*

Variable	Description
Term	top-level terms (in lowercase); terms that are parents are preceded by a plus (+) symbol.
Role	part of speech of the term, entity classification of the term, or the value Noun Group.
Attribute	attribute of the term.
Frequency	number of times the term appears in the document collection.
Number of Documents	number of documents in the collection in which the term appears.
Keep	**Y** if the term is used in subsequent nodes of the text mining analysis; **N** otherwise.
Parent/Child Status	plus (+) symbol if the term is a parent; blank otherwise.
Parent ID	key value of the term's parent. *Note:* Note: Each SAS Text Miner node outputs a terms data set, which is stored with the project data. The names of these data sets follow the format "NodeID_terms" (for example, TextParsing_terms). You can determine which term corresponds to a particular key by looking at this data set. For more information about where to find project data, see the "Opening SAS Enterprise Miner 4.x and 5.3 projects in SAS Enterprise Miner 12.1" topic in the SAS Enterprise Miner Help.
Rank	rank that corresponds to the ZIPF Plot.

For more information see the following:

- “Term Roles and Attributes” on page 36

Text Parsing Node SAS Output Results

The SAS output from the **Text Parsing** node includes summary information about the input variables.

Text Parsing Node Output Data

For information about output data for the **Text Parsing** node, see “Output Data for SAS Text Miner Nodes” on page 127.

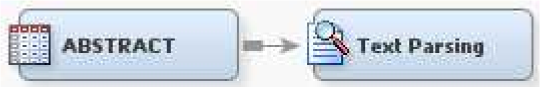
Using the Text Parsing Node

This example shows you how to identify terms and their instances in a data set containing text using the **Text Parsing** node. This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*. Perform the following steps:

1. The SAS data set SAMPSIO.ABSTRACT contains the titles and text of abstracts from conferences. Create the ABSTRACT data source and add it to your diagram workspace. Set the Role value of the TEXT and TITLE variables to `Text`.

2. Select the **Text Mining** tab on the toolbar, and drag a **Text Parsing** node into the diagram workspace.

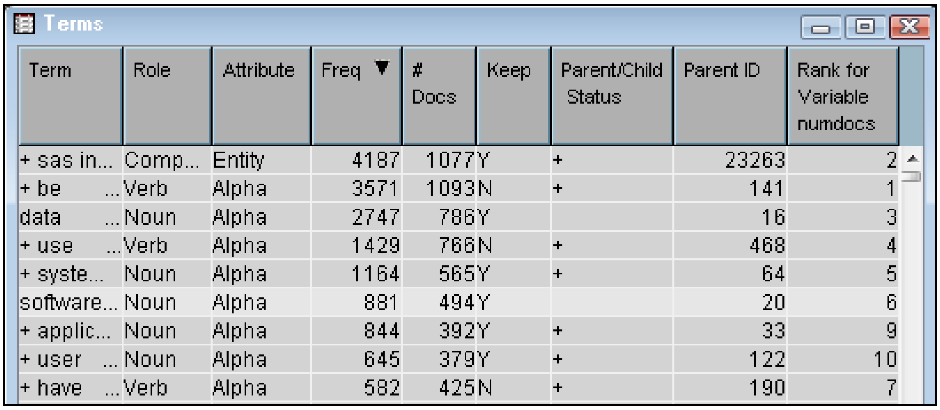
3. Connect the ABSTRACT data source to the **Text Parsing** node.



4. In the diagram workspace, right-click the **Text Parsing** node and select **Run**. Click **Yes** in the Confirmation dialog box that appears.

5. Click **Results** in the Run Status dialog box when the node finishes running. The Results window displays a variety of tabular and graphical output to help you analyze the terms and their instances in the ABSTRACT data source.

6. Sort the terms in the Terms table by frequency, and then select the term “software.” As the Terms table illustrates, the term “software” is a noun that occurs in 494 documents in the ABSTRACT data source, and appears a total number of 881 times.

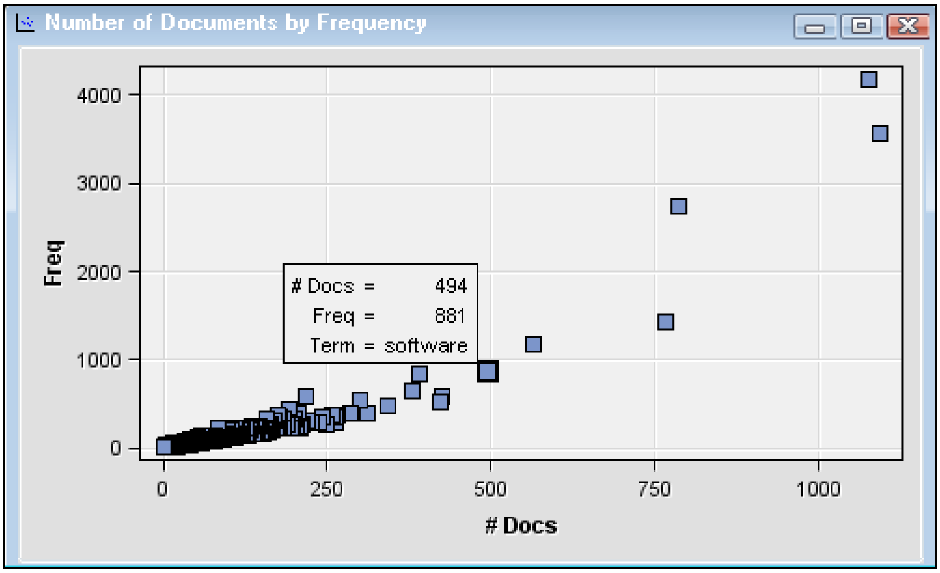


Terms

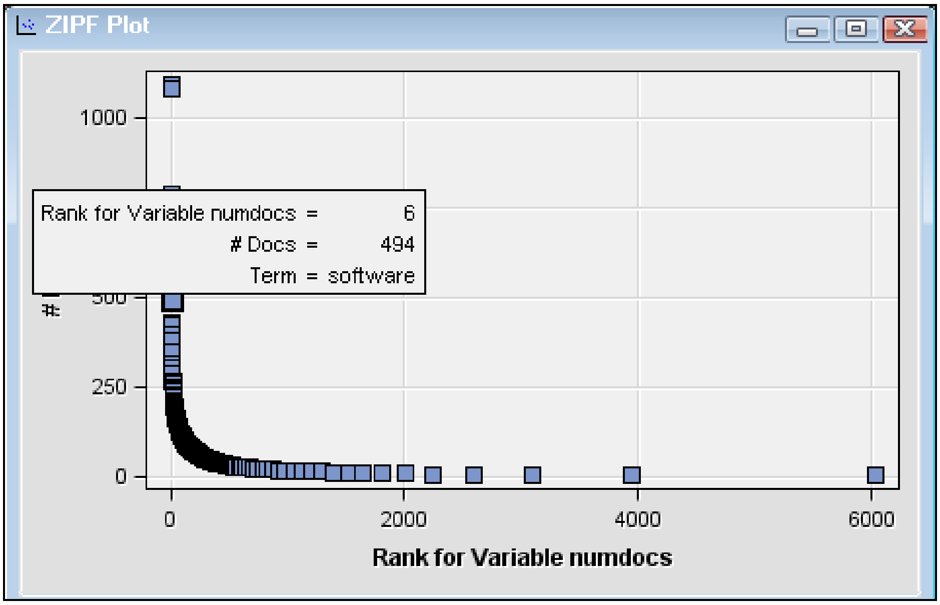
Term	Role	Attribute	Freq ▼	# Docs	Keep	Parent/Child Status	Parent ID	Rank for Variable numdocs
+ sas in...	Comp...	Entity	4187	1077	Y	+	23263	2
+ be ...	Verb	Alpha	3571	1093	N	+	141	1
data ...	Noun	Alpha	2747	786	Y		16	3
+ use ...	Verb	Alpha	1429	766	N	+	468	4
+ syste...	Noun	Alpha	1164	565	Y	+	64	5
software...	Noun	Alpha	881	494	Y		20	6
+ applic...	Noun	Alpha	844	392	Y	+	33	9
+ user ...	Noun	Alpha	645	379	Y	+	122	10
+ have ...	Verb	Alpha	582	425	N	+	190	7

When you select a term in the Terms table, the point corresponding to that term in the Text Parsing Results plots is highlighted.

7. Select the Number of Documents by Frequency plot, and position the cursor over the highlighted point for information about the term “software.”

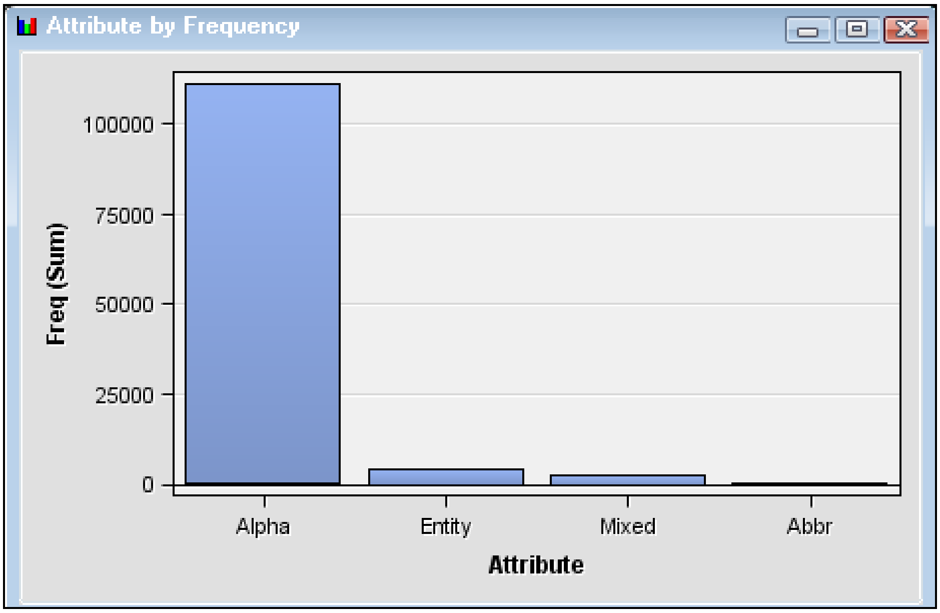


Similar information is also presented in a ZIPF plot.

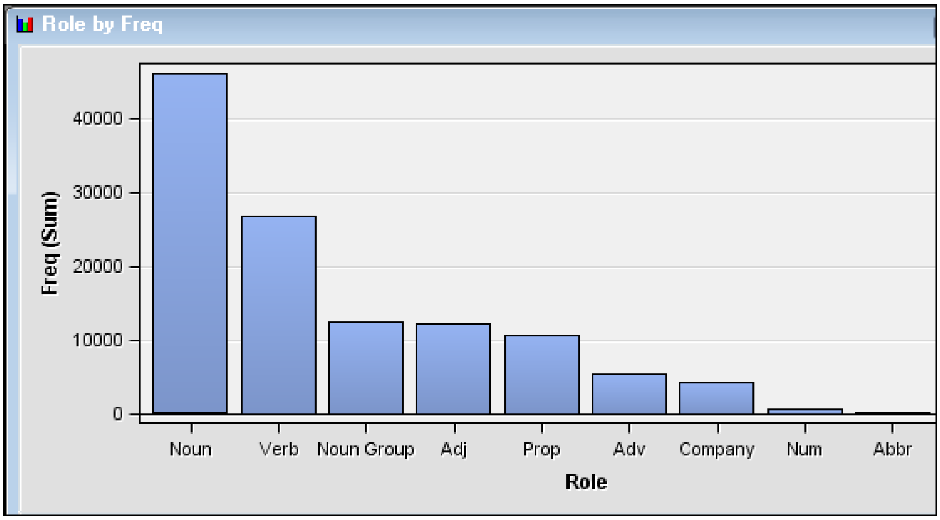


```
Rank for Variable numdocs =        6
                   # Docs =      494
                     Term = software
```

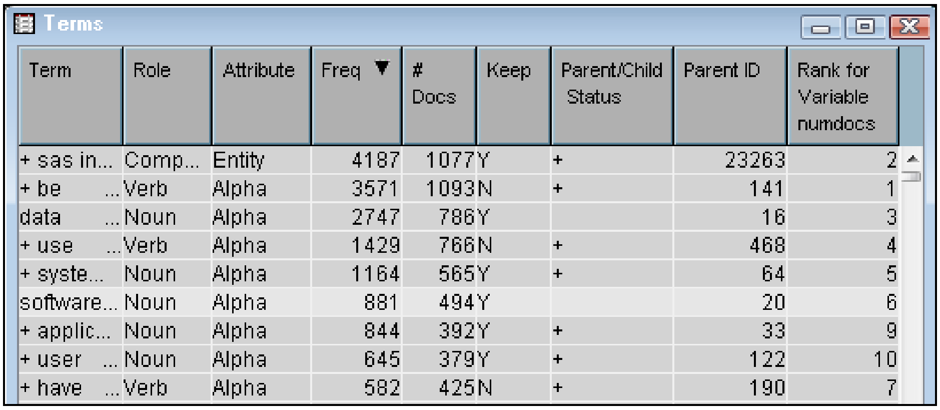
The Attribute by Frequency chart shows that **`Alpha`** has the highest frequency among attributes in the document collection.



The Role by Freq chart illustrates that **`Noun`** has the highest frequency among roles in the document collection.



8. Return to the Terms table, and notice that the term “software” is kept in the text parsing analysis. This is illustrated by the value of `Y` in the Keep column. Notice that not all terms are kept when you run the **Text Parsing** node with default settings.



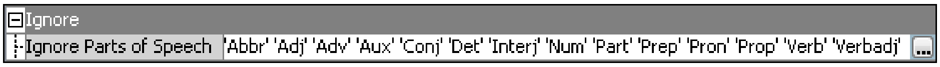
Term	Role	Attribute	Freq ▼	# Docs	Keep	Parent/Child Status	Parent ID	Rank for Variable numdocs
+ sas in...	Comp...	Entity	4187	1077	Y	+	23263	2
+ be ...	Verb	Alpha	3571	1093	N	+	141	1
data ...	Noun	Alpha	2747	786	Y		16	3
+ use ...	Verb	Alpha	1429	766	N	+	468	4
+ syste...	Noun	Alpha	1164	565	Y	+	64	5
software...	Noun	Alpha	881	494	Y		20	6
+ applic...	Noun	Alpha	844	392	Y	+	33	9
+ user ...	Noun	Alpha	645	379	Y	+	122	10
+ have ...	Verb	Alpha	582	425	N	+	190	7

The **Text Parsing** node not only enables you to gather statistical data about the terms in a document collection, but it also enables you to modify your output set of parsed terms by dropping terms that are a certain part of speech, type of entity, or attribute. Scroll down the list of terms in the Terms table and notice that many of the terms with a role other than `Noun` are kept. Let us assume that we want to limit our text parsing results to terms with a role of `Noun`.

9. Close the Results window.

10. Select the **Text Parsing** node, and then select the ellipsis for the **Ignore Parts of Speech** property.

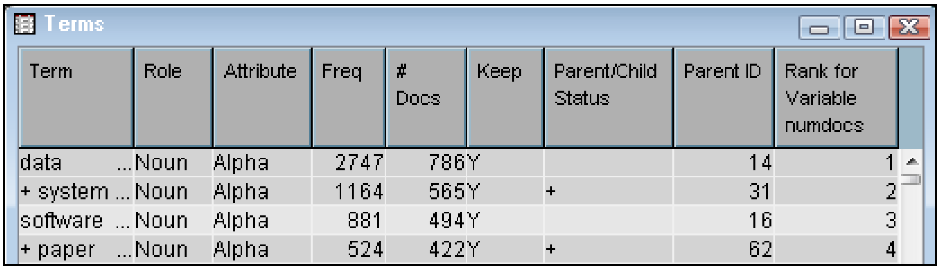
11. In the Ignore Parts of Speech dialog box, select all parts of speech except for `Noun` by holding Ctrl down on your keyboard and clicking on each option. Click **OK**. Notice that the value for the **Ignore Parts of Speech** property is updated with your selection.



⊟Ignore
Ignore Parts of Speech 'Abbr' 'Adj' 'Adv' 'Aux' 'Conj' 'Det' 'Interj' 'Num' 'Part' 'Prep' 'Pron' 'Prop' 'Verb' 'Verbadj' ...

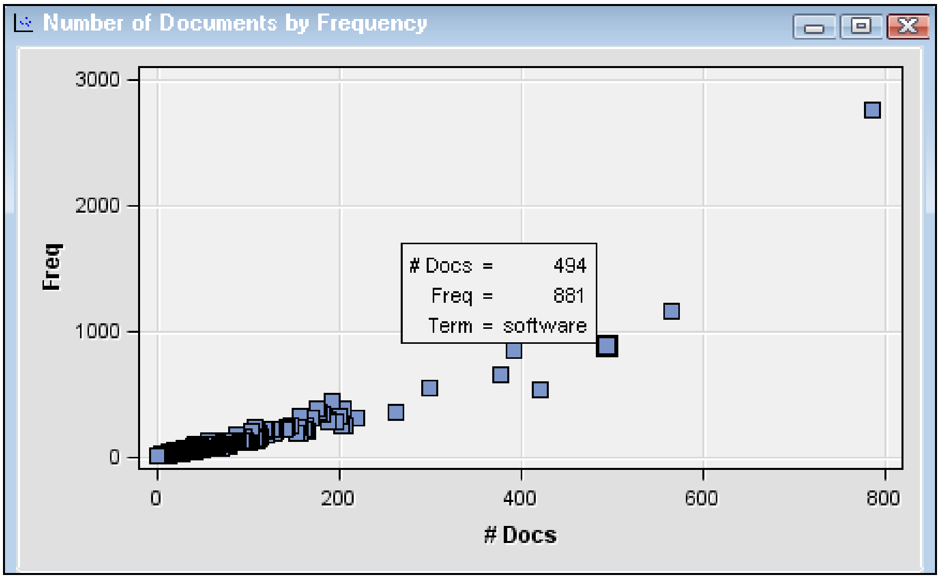
12. In addition to nouns, let us also keep noun groups. Set the **Noun Groups** property to `Yes`.

13. Right-click the **Text Parsing** node and select **Run**. Click **Yes** in the Confirmation dialog box that appears. Select **Results** in the Run Status dialog box when the node has finished running. Notice that the term “software” has a higher rank among terms with a role of just “noun” or “noun group” than it did when other roles were included. If you scroll down in the Terms table, you can see that just terms with a **`Noun`** or **`Noun Group`** role are included.

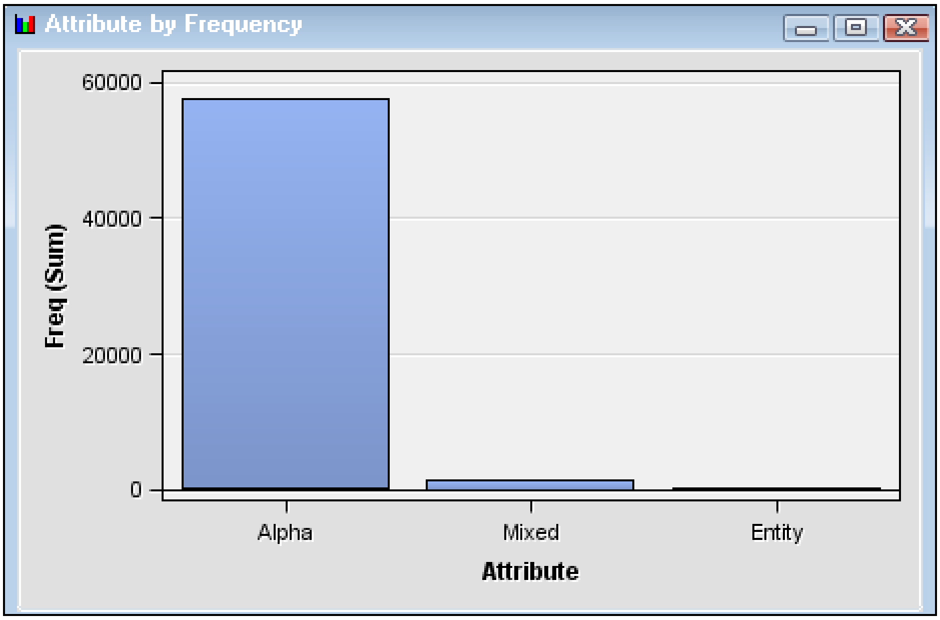


Term	Role	Attribute	Freq	# Docs	Keep	Parent/Child Status	Parent ID	Rank for Variable numdocs
data ...	Noun	Alpha	2747	786	Y		14	1
+ system ...	Noun	Alpha	1164	565	Y	+	31	2
software ...	Noun	Alpha	881	494	Y		16	3
+ paper ...	Noun	Alpha	524	422	Y	+	62	4

As we would expect, there are fewer terms plotted in the Number of Documents by Frequency plot:



Similarly, the total number of terms in the output results with an attribute of **`Alpha`** has decreased, as can be seen in the Attribute by Frequency chart:



Start Lists and Stop Lists

You can use a start list or a stop list in the **Text Parsing** node. These lists enable you to control which terms are or are not used in a text mining analysis. A “start list” is a data set that contains a list of terms to include in the parsing results. If you use a start list, then only terms that are included in that list appear in parsing results. A “stop list,” on the other hand, is a data set that contains a list of terms to exclude from the parsing results. Stop lists are often used to exclude terms that contain little information or that are extraneous to your text mining tasks. A default stop list is provided for each of several languages. For more information, see “SAS Text Miner Sample Data Sets” on page 9.

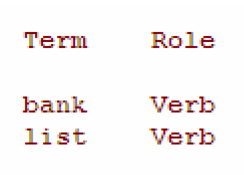
Start lists and stop lists have the same required format. You must include the variable “Term,” which contains the terms to include or exclude, respectively. In addition, you can include the variable “Role,” which contains an associated role. If you include “Role” and you have set the **Different Parts of Speech** property for the **Text Parsing** node to **Yes**, then terms are excluded or included based on the **(Term, Role)** pair.

Note: A role is either a part of speech, an entity classification, or the value **Noun Group**. For more information about roles, see “Term Roles and Attributes” on page 36.

For example, if you use the following stop list, then any instance of the terms “bank” and “list” are excluded from parsing results, regardless of their roles:

```
Term

bank
list
```

However, if you use the following stop list and the **Different Parts of Speech** property has the value `Yes`, then the terms “bank” and “list” are excluded from parsing results only if they are used as verbs:



```
Term    Role

bank    Verb
list    Verb
```

Term Stemming

Stemming is the process of finding the stem or root form of a term. SAS Text Miner uses dictionary-based stemming, which unlike tail-chopping stemmers, produces only valid words as stems. When part-of-speech tagging is on, the stem selection process restricts the stem to be of the same part-of-speech as the original term.

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

Stemming can be very important for text mining because text mining is based on the co-occurrence relationships of terms throughout the collection. By treating the variations of a term as the term itself, document relationships can be clarified. For example, if “grinds”, “grinding”, and “ground” each occur independently in three separate documents, the individual terms do not contribute to the similarity of these three documents. However, if the terms are all stemmed to “grind” and the documents are treated as if they contain “grind” rather than the original variants, the documents will be related by this common stem.

Because SAS Text Miner uses the same equivalent term concept to manage stems as it does to manage synonyms, you can customize the stem by editing the synonym list.

Furthermore, SAS Text Miner supports compound words for German. Compound words are decomposed only when stemming is performed. When they are detected, a parent form is generated that separates the compound into its parts by delimiting the parts with the # symbol. The individual components of the compound word are also added as terms. For example, the compound word “eckkonsole” is assigned the parent term “eck#konsole” and this parent as well as “eck” and “konsole” are added to the Terms table.

Other examples of compound words include “levensecht,” “kinderloos,” “naisjuoksija,” “Obstanbaugebiet,” and “hellgelb.”

Term Roles and Attributes

Contents

Parts of Speech in SAS Text Miner

SAS Text Miner can identify the part of speech for each term in a document based on the context of that term. Terms are identified as one of the following parts of speech:

- Abbr (abbreviation)
- Adj (adjective)
- Adv (adverb)
- Aux (auxiliary or modal)
- Conj (conjunction)
- Det (determiner)
- Interj (interjection)
- Noun (noun)
- Num (number or numeric expression)
- Part (infinitive marker, negative participle, or possessive marker)
- Prep (preposition)
- Pron (pronoun)
- Prop (proper noun)
- Punct (punctuation)
- Verb (verb)
- VerbAdj (verb adjective)

Noun Groups in SAS Text Miner

SAS Text Miner can identify noun groups, like “clinical trial” and “data set”, in a document collection. Noun groups are identified based on linguistic relationships that exist within sentences. Syntactically, these noun groups act as single units and you can, therefore, choose to parse them as single terms.

- If stemming is on, noun groups are stemmed. For example, the text “amount of defects” is parsed as “amount of defect”.
- Frequently, shorter noun groups are contained within larger noun groups; both the shorter and larger noun groups appear in parsing results.

Entities in SAS Text Miner

An “entity” is any of several types of information that SAS Text Miner can distinguish from general text. If you enable SAS Text Miner to identify them, entities are analyzed as a unit, and they are sometimes normalized. When SAS Text Miner extracts entities that consist of two or more words, the individual words of the entity are also used in the analysis.

Out of the box, SAS Text Miner identifies the following standard entities:

- ADDRESS (postal address or number and street name)
- COMPANY (company name)
- CURRENCY (currency or currency expression)
- DATE (date, day, month, or year)
- INTERNET (e-mail address or URL)
- LOCATION (city, country, state, geographical place or region, or political place or region)
- MEASURE (measurement or measurement expression)
- ORGANIZATION (government, legal, or service agency)
- PERCENT (percentage or percentage expression)
- PERSON (person's name)
- PHONE (phone number)
- PROP_MISC (proper noun with an ambiguous classification)
- SSN (Social Security number)
- TIME (time or time expression)
- TIME_PERIOD (measure of time expressions)
- TITLE (person's title or position)
- VEHICLE (motor vehicle including color, year, make, and model)

You can also use SAS Concept Creation for SAS Text Miner or SAS Content Categorization Studio to define custom entities and import these for use in a **Text Parsing** node. When you create compiled custom entity files (valid files have the extension .li), ensure that you specify March 29, 2012, as the compatibility date. Otherwise, the files cannot be used in SAS Text Miner.

Entities are normalized in these situations:

- SAS Text Miner uses a fixed dictionary of company and organization names in order to identify these entity types, and entities of this type will frequently be associated with a parent. For example, if "IBM" appears in the text, it is returned with the predefined parent "International Business Machines". Typically, the longest and most precise version of a name is used as the parent form.
- SAS Text Miner normalizes entities that have an ISO (International Standards Organization) standard (dates or years, currencies, and percentages). Rather than return the normalization as a parent of the original term, these normalizations actually replace the original term.
- You can alter any parent forms that are returned by editing the synonym list. Place terms that you want to identify as an entity in the "Term" variable, place the parent to associate with it in the "Parent" variable, and place the entity category in the "Category" variable. Then rerun the node.

Attributes in SAS Text Miner

When a document collection is parsed, SAS Text Miner categorizes each term as one of the following attributes, which gives an indication of the characters that compose that term:

- Alpha, if characters are all letters
- Entity, if the term is an entity
- Num, if term characters include a number
- Punct, if the term is a punctuation character
- Mixed, if term characters include a mix of letters, punctuation, and white space
- Abbr, if the term is an abbreviation

Synonym Lists

Contents

Overview of Synonym Lists

You can use a synonym list in the **Text Parsing** node. A synonym list enables you to specify different words that should be processed equivalently, as the same representative parent term. A default synonym list is provided for the English language. For more information, see "SAS Text Miner Sample Data Sets" on page 9.

Synonym data sets have a required format. If your synonyms data set does not contain the variables listed below, then the **Text Parsing** node will return an error.

You must include the following variables:

- "Term" contains a term to treat as a synonym of "Parent."
- "Parent" contains the representative term to which "Term" should be assigned.
- "Category" enables you to specify that the synonym is assigned only when "Term" occurs with a specific role.

Note: If a synonym list includes multiple entries that assign the same terms to different parents, then the parsing results will reflect only the first entry.

"Category" enables you to specify that the same word, when it has different roles, can be processed either as different synonyms or as itself. For example, you can specify that the term "SAS," when tagged as a noun, is parsed as "SAS Institute" but, when tagged as a verb, is processed as "sas." In order for the variable "Category" to be used when the **Text Parsing** node processes a synonym list, the **Noun Groups** and **Different Parts of Speech** properties must be set to **`Yes`**, and the **Find Entities** property should not be set to **`None`**, if these roles are used in the synonym list.

You can use the following format, which enables you to change a part of speech tag.

Term	Termrole	Parent	Parentrole
car	Noun	vehicle	Noun

You can also use the following format. For example, use of the following synonym list causes any instance of "SAS," when identified as a company, to be processed as "SAS Institute." Also, any instance of "employees" is processed as "employees." In fact, if part-of-speech tagging is on, then this entry disables stemming for only the term "employees."

Term	Parent	Category
SAS	SAS Institute	Company
employees	employees	

Synonyms and Part-of-Speech Tagging

The following examples demonstrate how synonym lists are handled when part-of-speech tagging is on.

Term	Parent	Category
well	water	

In this example, a term and parent (but not a category) are defined. When part-of-speech tagging is either on or off, every occurrence of “well,” regardless of part of speech, is assigned to the parent “water.”

Term	Parent	Category
well		noun
data mining		noun

In this example, a term and category (but not a parent) are defined. When part-of-speech tagging is either on or off, the entry for the single word term, “well,” has no effect on the parsing results. However, when part-of-speech tagging is on, the multi-word term, “data mining” is treated as a single term only when identified as a noun. When part-of-speech tagging is off, any instance of “data mining” is treated as a single term.

Term	Parent	Category
well	water	noun

In this example, a term, parent, and category are defined. When part-of-speech tagging is on, “well” is assigned to the parent “water” only when it is identified as a noun. When part-of-speech tagging is off, all instances of “well” are assigned to the parent “water.”

Defining Multi-Word Terms Using a Synonym List

You can use a synonym to specify groups of words that should be processed together as single terms. To define a multi-word term, include it as a “Term” in a synonym list; do not assign it to a “Parent.” For more information, see “Multi-Term Lists” on page 41.

Unlike other entries in a synonym list, multi-word terms are case-sensitive. Appearances of the multi-word term that have the following casings are identified and treated as a single term:

- same casing as the multi-word term entry in the synonym list
- all uppercase version of the multi-word term
- all lowercase version of the multi-word term
- a version that capitalizes the first letter of each term in the multi-word term and lowercases the remaining characters

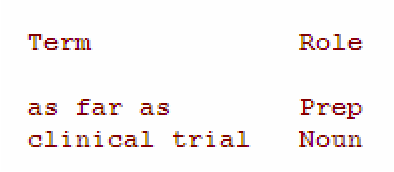
Multi-Term Lists

You can use a multi-term list in the **Text Parsing** node. These lists enable you to specify groups of words that should be processed together, as single terms. A default multi-term list is provided for each of several languages. For more information, see “SAS Text Miner Sample Data Sets” on page 9.

Multi-word term data sets have a required format. You must include the variables “Term,” which contains a multi-word term, and “Role,” which contains an associated role.

Note: A role is either a part of speech, an entity classification, or the value **Noun Group**. For more information about roles, see “Term Roles and Attributes” on page 36.

For example, if you use the following multi-term list, then any instance of the words “as far as” is processed as one term, a preposition, and any instance of the words “clinical trial” is processed as one term, a noun:



```
Term              Role

as far as         Prep
clinical trial    Noun
```

You can similarly define multi-word terms using a synonym list. In this case, the groups of words that you specify will be processed together as single terms. For more information, see “Defining Multi-Word Terms Using a Synonym List” on page 40.

Chapter 5
The Text Filter Node

Overview of the Text Filter Node

You can use the **Text Filter** node to reduce the total number of parsed terms or documents that are analyzed. Therefore, you can eliminate extraneous information so that only the most valuable and relevant information is considered. For example, the **Text Filter** node can be used to remove unwanted terms and to keep only documents that discuss a particular issue. This reduced data set can be orders of magnitude smaller than the one representing the original collection that might contain hundreds of thousands of documents and hundreds of thousands of distinct terms. See the following for more information about the **Text Filter** node.

- "Text Filter Node Input Data" on page 44
- "Text Filter Node Properties" on page 45
- "Text Filter Node Results" on page 48
- "Text Filter Node Output Data" on page 51
- "Using the Text Filter Node" on page 51

For related topics that you might find useful when using the **Text Filter** node, see the following:

- "How to Create a Dictionary Data Set" on page 56
- "Term Weighting" on page 57
- "Text Filter Node Search Expressions" on page 59
- "Interactive Filter Viewer" on page 60
- "Strength of Association for Concept Linking" on page 63

Note: The **Text Filter** node is not supported for use in group processing (Start Groups and End Groups nodes).

Text Filter Node Input Data

The **Text Filter** node must be preceded by a **Text Parsing** node in a process flow diagram. The **Text Filter** node directly imports the document table from the **Text Parsing** node, but it also relies on several data sets that the **Text Parsing** node places in its workspace directory.

Text Filter Node Properties

Contents

Text Filter Node General Properties

These are the general properties that are available on the **Text Filter** node:

- **Node ID** — displays the ID that is assigned to the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Filter** node that is added to a diagram will have the Node ID **`TextFilter`**, and the second **Text Filter** node added will have the Node ID **`TextFilter2`**.

- **Imported Data** — accesses a list of the data sets that are imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set

- **Exported Data** — accesses a list of the data sets that are exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set

- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Filter Node Train Properties

General Train Properties

These are the training properties that are available on the **Text Filter** node:

- **Variables** — accesses a list of variables and associated properties in the data source. Click the ellipsis button to open the Variables window.
- "Spelling Properties" on page 46
- "Weightings Properties" on page 46
- "Term Filters Properties" on page 46
- "Document Filters Properties" on page 47
- "Results Properties" on page 47

Spelling Properties

- **Check Spelling** — specifies whether to check spelling and create synonyms for misspelled words.
- **Dictionary** — specifies a data set of correctly spelled terms. Click the ellipsis button to open the Select a SAS Table window, and select a dictionary data set to use during spell-checking. For more information about how to create a dictionary data set, see "How to Create a Dictionary Data Set" on page 56.

  Note: The **Check Spelling** property value must be set to `Yes` to use the dictionary data set.

Weightings Properties

- **Frequency Weighting** — specifies the frequency weighting method to use. For more information, see "Frequency Weighting Methods" on page 58.
- **Term Weight** — specifies the term weighting method to use. For more information, see "Term Weighting Methods" on page 58.

Term Filters Properties

- **Minimum Number of Documents** — excludes terms that occur in fewer than this number of documents.
- **Maximum Number of Terms** — specifies the maximum number of terms to keep.
- **Import Synonyms** — specifies a synonym data set to import. Click the ellipsis button to browse to a data set. Synonym data sets should have the variables term and parent. The variables termrole and parentrole are optional variables that give you control of changing term role assignments (part-of-speech tag or entity category), if desired.

Document Filters Properties

- **Search Expression** — specifies a search expression to use to filter documents. For more information, see "Text Filter Node Search Expressions" on page 59.
- **Subset Documents** — accesses a window in which you can build a WHERE clause to use to filter documents. Only documents that satisfy this WHERE clause are kept. For more information about WHERE clauses and WHERE-expression processing, see *SAS 9.2 Language Reference: Concepts* at `http://support.sas.com/documentation/onlinedoc/base/index.html`. Click the ellipsis button to open the Build Where Clause dialog box, in which you can do the following:
  - Select from the drop-down menus to build a WHERE clause.
  - Enter the full text of a custom WHERE clause.

Results Properties

- **Filter Viewer** — (after the node has run) accesses the "Interactive Filter Viewer" on page 60, in which you can interactively refine the parsed and filtered data. Click the ellipsis button to open the Interactive Filter Viewer.
- **Spell-Checking Results** — (after the node has run and if the value of **Check Spelling** is `Yes`) accesses a window in which you can view the data set that contains spelling corrections generated during spell-checking. Click the ellipsis button to view the data set.

  In the spell-checking results data set, the variable term contains the proposed misspelled terms, and the variable parent contains the associated proposed correct spellings of these terms. If you want to edit this file, you can do so using SAS code.
- **Exported Synonyms** — opens a data set containing synonyms that are exported from the Interactive Filter Viewer. For more information about how to create a synonym data set from the Interactive Filter Viewer, see "Create Synonym Data Sets" on page 62.

Text Filter Node Report Properties

These are the report properties that are available on the **Text Filter** node:

- **Terms to View** — specifies which terms to display in the Results window.
  - **Selected** displays all terms that were kept after filtering.
  - **Filtered** displays all terms that were dropped after filtering.
  - **All** displays all terms both kept and dropped after filtering.
- **Number of Terms to Display** — indicates the maximum number of terms to be displayed in the Results viewer. Terms are first sorted by the number of documents in which they appear, and then the list is truncated to the maximum number. If the value of this property is `All`, then all terms are displayed.

Text Filter Node Status Properties

These are the status properties that are displayed on the **Text Filter** node:

- **Create Time** — time that the node was created.
- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.
- **Last Error** — error message, if any, from the last run.
- **Last Status** — last reported status of the node.
- **Last Run Time** — time at which the node was last run.
- **Run Duration** — length of time required to complete the last node run.
- **Grid Host** — grid host, if any, that was used for computation.
- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner Extension node. The value of this property is always **No** for the **Text Filter** node.

Text Filter Node Results

Contents

Results Window for the Text Filter Node

After the Text Filter node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status window that opens immediately after a successful run of the **Text Filter** node.
- Click the Results icon on the main Toolbar.
- Right-click the **Text Filter** node, and select **Results**.

The Results window for the **Text Filter** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are the same as other nodes. However, the **View** menu does not include the selections **Assessment** or **Model**. Instead, it includes **Terms** and **Filtering**, which access submenus that list the graphical and tabular results.

Note: You can access the SAS log from the **SAS Results** submenu of the **View** menu. This log can be a useful debugging tool.

Text Filter Node Graphical and Tabular Results

The following are the graphical results in the **Text Filter** node Results window:

- The **Number of Documents** scatter plot displays the imported number of documents in which a term appears versus the number of documents. Each data point represents a term. If you position the mouse pointer over a plotted point, then a tooltip indicates the term name, the imported number of documents for a term, the number of documents in which that term appears, and the weight of the term.

  Note: To generate the Number of Documents scatter plot, you need to set the **Check Spelling** property of the **Text Filter** node to `Yes` before running the node. In the Results Window, select **View** ⇨ **Filtering** ⇨ **Number of Documents**.

- The **Number of Documents by Weight** scatter plot displays the number of documents in which a term appears versus the weight of the term. Each data point represents a term. If you position the mouse pointer over a plotted point, then a tooltip indicates the term name, the number of documents in which that term appears, and the weight of the term. For more information, see "Term Weighting" on page 57.

- The **Number of Documents by Frequency** scatter plot displays the number of documents in which a term appears versus the frequency of occurrence of that term in the entire document collection. Each data point represents a term. If you position the mouse pointer over a plotted point, then a tooltip indicates the term name, the number of documents in which that term appears, and the number of times that term appears in the entire document collection.

- The **Role by Freq** bar chart displays the total frequency of occurrence of terms in the document collection, broken down by term role and keep status. Each bar represents a role. If you position the mouse pointer over a bar, then a tooltip indicates the role name, the number of times a term with that role appears in the entire document collection, and the keep status of the associated terms.

- The **Attribute by Frequency** bar chart displays the total frequency of occurrence of terms in the document collection, broken down by attribute and keep status. If you position the mouse pointer over a bar, then a tooltip indicates the attribute name, the number of times a term with that attribute appears in the entire document collection, and the keep status of the associated terms.

- The **ZIPF Plot** displays a scatter plot of the number of documents for each term where each is sorted and plotted by rank. If you position the mouse pointer over a point, then a tooltip indicates the term name, rank, and number of documents.

There are three tabular results in the **Text Filter** node Results window:

- The **Terms** table displays information about top-level terms (in other words, terms that have no parents above them). All graphical results in the **Text Filter** node Results window are linked to this table. Therefore, you can select an observation in the Terms table and the associated data points are highlighted in the graphics. Or, you can select data points in the graphics and the associated observations are highlighted in the Terms table.

- The **Excluded Terms** table (accessible via the **Filtering** submenu of the **View** menu) displays information about all dropped terms.

- The **New Parent Terms** table (accessible via the **Filtering** submenu of the **View** menu) displays information about all terms that were newly classified as parent terms in the **Text Filter** node.

Table 5.1 Contents of the Terms Table

Variable	Description
Term	top-level terms (in lowercase); terms that are parents are preceded by a plus (+) symbol.
Role	part of speech of the term, entity classification of the term, or the value **`Noun Group`**.
Attribute	attribute of the term.
Status	**`Keep`** if the term is used in subsequent nodes of the text mining analysis; **`Drop`** otherwise.
Weight	weight of the term
Imported Frequency	number of times the term appears in the document collection; this variable reflects the frequency passed from the previous node.
Frequency	number of times the term appears in the document collection; this variable reflects the frequency after filtering.
Number of Imported Documents	number of documents in the collection in which the term appears; this variable reflects the number passed from the previous node.
Number of Documents	number of documents in the collection in which the term appears; this variable reflects the number after filtering.
Rank	rank that corresponds to the ZIPF Plot.
Parent/Child Status	plus (+) symbol if the term is a parent; blank otherwise.

Variable	Description
Parent ID	key value of the term's parent. *Note:* Each SAS Text Miner node outputs a terms data set, which is stored with the project data. The names of these data sets follow the format “NodeID_terms” (for example, TextFilter_terms). You can determine which term corresponds to a particular key by looking at this data set. For more information about where to find project data, see the “Opening SAS Enterprise Miner 4.x and 5.3 projects in SAS Enterprise Miner 12.1” topic in the SAS Enterprise Miner Help.

For more information, see:

- “Term Roles and Attributes” on page 36
- “Term Weighting” on page 57

Text Filter Node SAS Output Results

The SAS output from the **Text Filter** node includes summary information about the input variables.

Text Filter Node Output Data

For information about output data for the **Text Filter** node, see “Output Data for SAS Text Miner Nodes” on page 127.

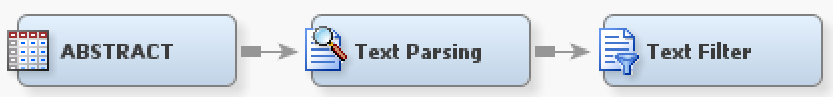
Using the Text Filter Node

This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

The **Text Filter** node enables you to reduce the total number of terms in your text mining analysis. For example, common or infrequent words might not be useful to analyze, and can be filtered out. This example shows you how to filter out terms using the **Text Filter** node. This example assumes that you have performed “Using the Text Parsing Node” on page 29, and builds off the process flow diagram created there.

1. Select the **Text Mining** tab on the toolbar, and drag a **Text Filter** node into the diagram workspace.

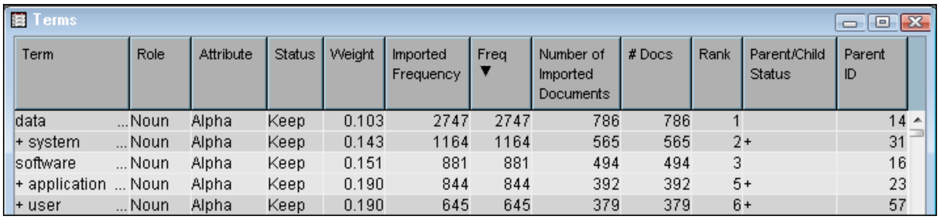
2. Connect the **Text Parsing** node to the **Text Filter** node.



3. In the diagram workspace, right-click the **Text Filter** node and select **Run**. Click **Yes** in the Confirmation dialog box.

4. Click **Results** in the Run Status dialog box when the node finishes running.

5. Select the Terms table. Sort the terms by frequency by clicking the Freq column heading.

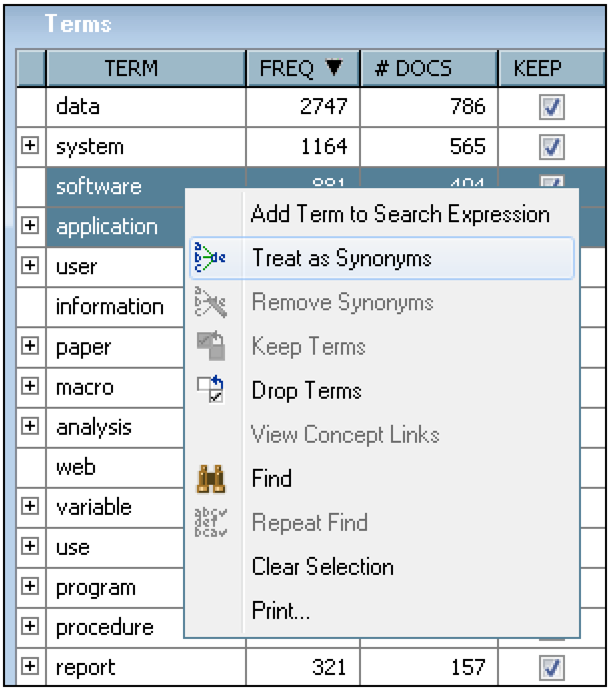


Term	Role	Attribute	Status	Weight	Imported Frequency	Freq ▼	Number of Imported Documents	# Docs	Rank	Parent/Child Status	Parent ID
data	...Noun	Alpha	Keep	0.103	2747	2747	786	786	1		14
+ system	...Noun	Alpha	Keep	0.143	1164	1164	565	565	2	+	31
software	...Noun	Alpha	Keep	0.151	881	881	494	494	3		16
+ application	...Noun	Alpha	Keep	0.190	844	844	392	392	5	+	23
+ user	...Noun	Alpha	Keep	0.190	645	645	379	379	6	+	57

Assume that for purposes of our text mining analysis, we know that "software" and "application" are really used as synonyms in the documents that we want to analyze, and we want to treat them as the same term.

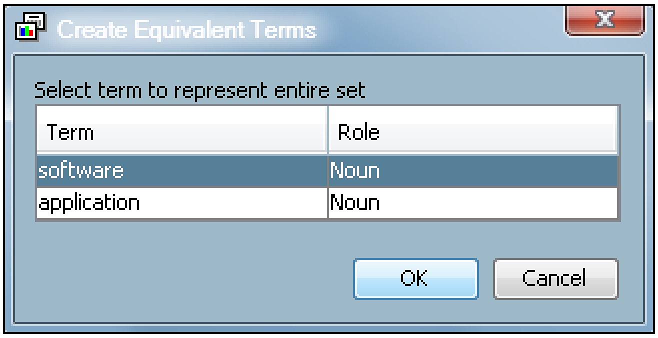
6. Close the Results window. Select the **Text Filter** node, and then click the ellipsis button for the **Filter Viewer** property.

7. In the Interactive Filter Viewer sort the terms in the Terms table by frequency. Hold Ctrl down on your keyboard, select "software" and "application", and then right-click "software" and select **Treat as Synonyms** from the drop-down menu.



	TERM	FREQ ▼	# DOCS	KEEP
	data	2747	786	☑
⊞	system	1164	565	☑
	software			
⊞	application			
⊞	user			
	information			
⊞	paper			
⊞	macro			
⊞	analysis			
	web			
⊞	variable			
⊞	use			
⊞	program			
⊞	procedure			
⊞	report	321	157	☑

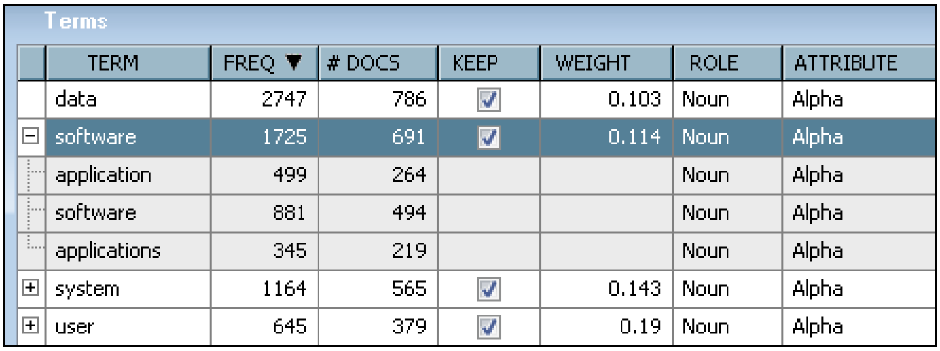
8. In the Create Equivalent Terms dialog box, select **software** as the term to represent both terms in the Terms table.



Select term to represent entire set

Term	Role
software	Noun
application	Noun

9. Click **OK** in the Create Equivalent Terms dialog box. Notice that the term “software” now represents both terms in the Terms table. Expand the term “software”.

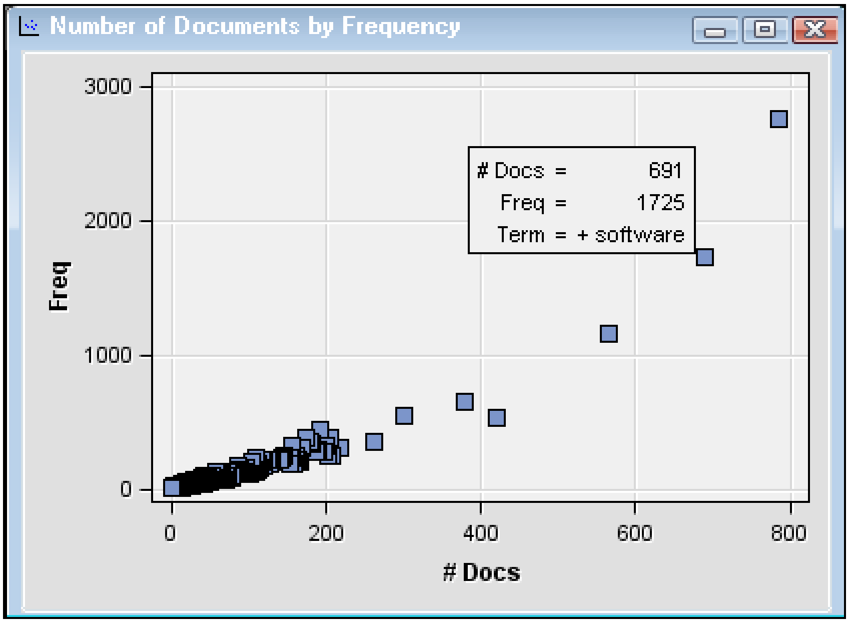


	TERM	FREQ ▼	# DOCS	KEEP	WEIGHT	ROLE	ATTRIBUTE
	data	2747	786	☑	0.103	Noun	Alpha
⊟	software	1725	691	☑	0.114	Noun	Alpha
	application	499	264			Noun	Alpha
	software	881	494			Noun	Alpha
	applications	345	219			Noun	Alpha
⊞	system	1164	565	☑	0.143	Noun	Alpha
⊞	user	645	379	☑	0.19	Noun	Alpha

10. Close the Interactive Filter Viewer. When prompted whether you would like to save your changes, select **Yes**.

11. Right-click the **Text Filter** node, and select **Run**. Select **Yes** in the Confirmation dialog box. Select **Results** in the Run Status dialog box when the node has finished running.

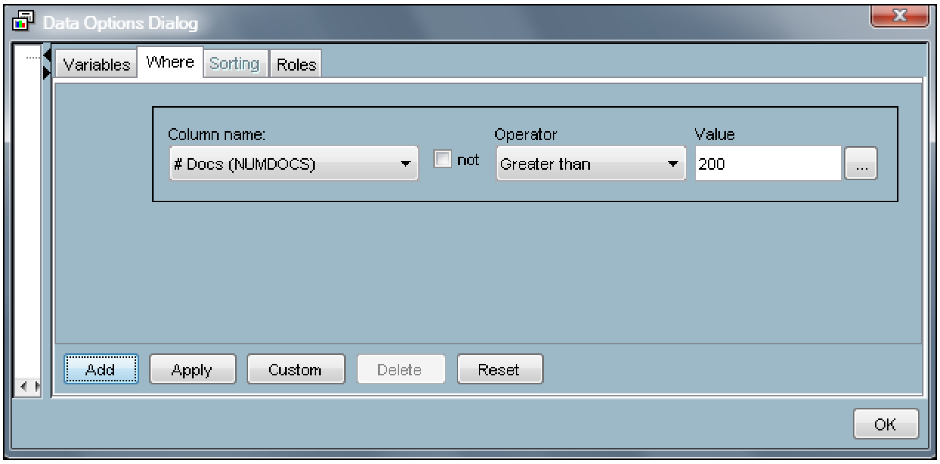
12. Select the Number of Documents by Frequency plot to see how both terms are now treated as the same.



You can also use options to change your view or specify a subset of results to appear in a plot. For example, consider that you want to refine this plot to only show terms that appear in more than 200 documents.

13. Right-click the Number of Documents by Frequency plot, and select **Data Options**.

14. Select the **Where** tab in the Data Options Dialog box. Select `# Docs` from the **Column name** drop-down menu. Select `Greater than` from the **Operator** drop-down menu. Type *200* in the **Value** text box.



Data Options Dialog

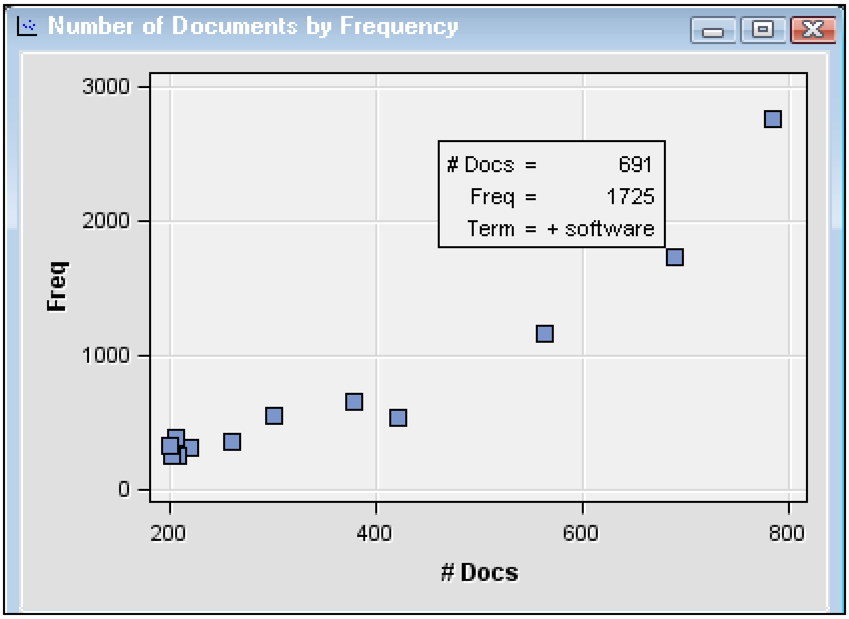
Column name: # Docs (NUMDOCS)

□ not

Operator: Greater than

Value: 200

15. Click **Apply**, and then click **OK**. The Number of Documents by Frequency plot resizes and includes only terms that occur in more than 200 documents.

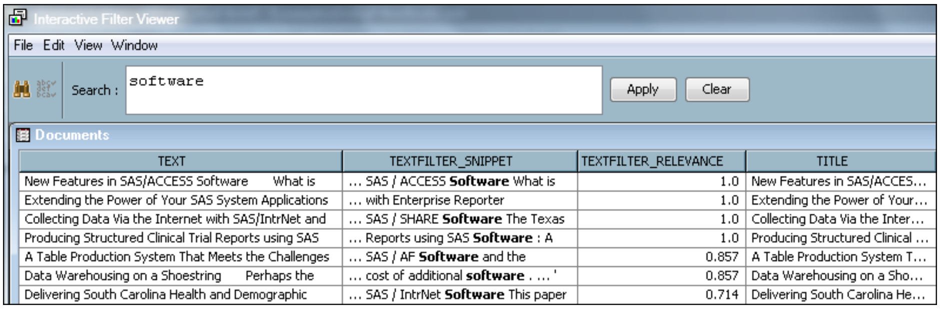


Number of Documents by Frequency

# Docs = 691
Freq = 1725
Term = + software

16. Close the Results window. In addition to resizing or subsetting a plot to help focus your analysis, you can also directly search for terms using the Interactive Filter Viewer.

17. Select the **Text Filter** node, and then click the ellipsis button for the **Filter Viewer** property. In the Interactive Filter Viewer, type *software* in the **Search** text box, and click **Apply**.



Search : software

TEXT	TEXTFILTER_SNIPPET	TEXTFILTER_RELEVANCE	TITLE
New Features in SAS/ACCESS Software What is	... SAS / ACCESS **Software** What is	1.0	New Features in SAS/ACCES...
Extending the Power of Your SAS System Applications	... with Enterprise Reporter	1.0	Extending the Power of Your...
Collecting Data Via the Internet with SAS/IntrNet and	... SAS / SHARE **Software** The Texas	1.0	Collecting Data Via the Inter...
Producing Structured Clinical Trial Reports using SAS	... Reports using SAS **Software** : A	1.0	Producing Structured Clinical ...
A Table Production System That Meets the Challenges	... SAS / AF **Software** and the	0.857	A Table Production System T...
Data Warehousing on a Shoestring Perhaps the	... cost of additional **software** . ... '	0.857	Data Warehousing on a Sho...
Delivering South Carolina Health and Demographic	... SAS / IntrNet **Software** This paper	0.714	Delivering South Carolina He...

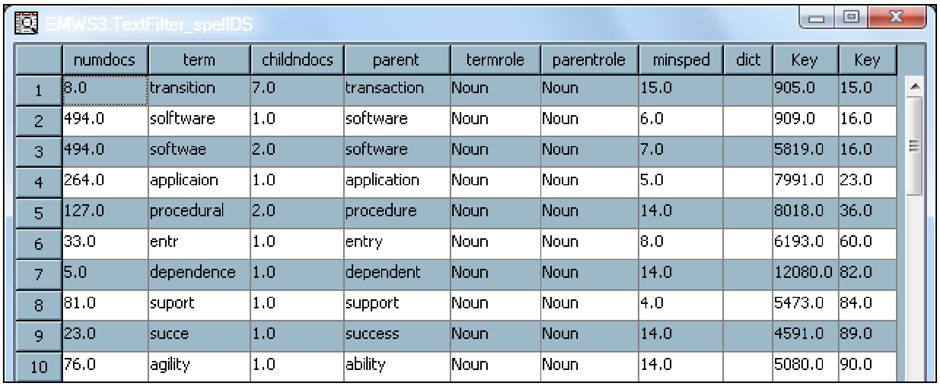
The Documents table provides a snippet of text that includes the term that you are searching for. You can use information in the Documents table to help you understand the context in which a term is being used by examining the snippet result in addition to the full text and title of the document. For more information about the Interactive Filter Viewer, see "Interactive Filter Viewer" on page 60.

Searching for a term in the Interactive Filter Viewer raises an interesting problem. As shown above, a search for "software" is case insensitive. However, what if there are instances of a term that we want to find that are misspelled in the document collection? You can also check for spelling when filtering terms using a dictionary data set.

18. Close the Interactive Filter Viewer, and select **No** when prompted for whether you want to save changes.

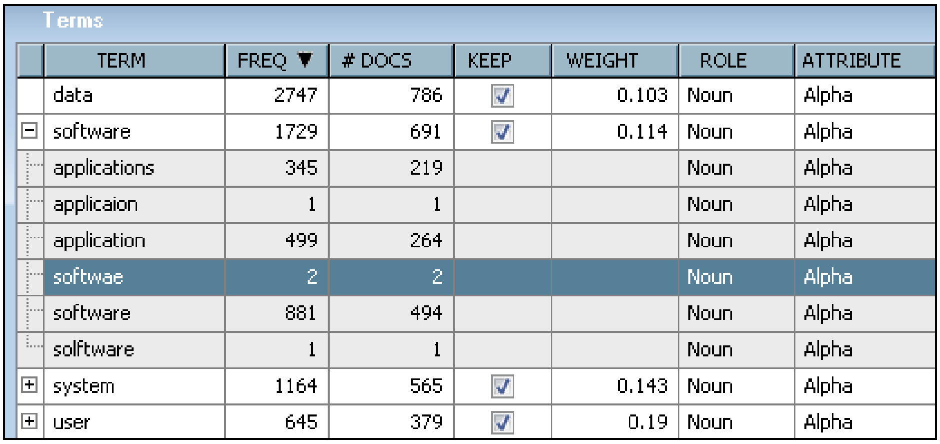
19. (Optional) Select the **Text Filter** node, and set the **Check Spelling** property to `Yes`. When you rerun the **Text Filter** node, terms will be checked for misspellings. You can also specify a data set to use in spell-checking by clicking the ellipsis button for the **Dictionary** property and selecting a data set. For information about creating a dictionary data set, see "How to Create a Dictionary Data Set" on page 56.

    Right-click the **Text Filter** node, and select **Run**. Select **Yes** in the Confirmation dialog box. When the node finishes running, select **OK** in the Run Status dialog box. Click the ellipsis button for the **Spell-Checking Results** property to access a window in which you can view the data set that contains spelling corrections generated during spell-checking. For example, the term "softwae" is identified as a misspelling of the term "software."



	numdocs	term	childndocs	parent	termrole	parentrole	minsped	dict	Key	Key
1	8.0	transition	7.0	transaction	Noun	Noun	15.0		905.0	15.0
2	494.0	solftware	1.0	software	Noun	Noun	6.0		909.0	16.0
3	494.0	softwae	2.0	software	Noun	Noun	7.0		5819.0	16.0
4	264.0	applicaion	1.0	application	Noun	Noun	5.0		7991.0	23.0
5	127.0	procedural	2.0	procedure	Noun	Noun	14.0		8018.0	36.0
6	33.0	entr	1.0	entry	Noun	Noun	8.0		6193.0	60.0
7	5.0	dependence	1.0	dependent	Noun	Noun	14.0		12080.0	82.0
8	81.0	suport	1.0	support	Noun	Noun	4.0		5473.0	84.0
9	23.0	succe	1.0	success	Noun	Noun	14.0		4591.0	89.0
10	76.0	agility	1.0	ability	Noun	Noun	14.0		5080.0	90.0

You can see this relationship in the Terms table in the Interactive Filter Viewer. Click the ellipsis button for the **Filter Viewer** property. Expand the term "software" in the Terms table to view its synonyms. The synonyms include "softwae," which was identified as a misspelled term during spell-checking.



Terms

	TERM	FREQ ▼	# DOCS	KEEP	WEIGHT	ROLE	ATTRIBUTE
	data	2747	786	☑	0.103	Noun	Alpha
⊟	software	1729	691	☑	0.114	Noun	Alpha
	applications	345	219			Noun	Alpha
	applicaion	1	1			Noun	Alpha
	application	499	264			Noun	Alpha
	softwae	2	2			Noun	Alpha
	software	881	494			Noun	Alpha
	solftware	1	1			Noun	Alpha
⊞	system	1164	565	☑	0.143	Noun	Alpha
⊞	user	645	379	☑	0.19	Noun	Alpha

Notice that the synonyms also include "application," which was created in steps 7-10 of this example, and "applicaion," which was identified during spell-checking as a misspelling of "application."

How to Create a Dictionary Data Set

You can use a “dictionary data set” as input to the **Text Filter** node or the %TEXTSYN macro for any language that is supported by SAS Text Miner.

Typically, if you specify to use spell checking during filtering, the words in a document collection are checked against each other and candidate misspellings are proposed. Including a dictionary data set for spell checking can reduce the number of misspellings that are falsely identified by this process. All words in the dictionary data set are viewed as correctly spelled, regardless of how they compare to other words in the collection.

There are several free dictionary sources that you can use with a DATA step to create a dictionary data set for use with SAS Text Miner. For example, OpenOffice.org has links to dictionaries for many languages that are available for free download from `http://extensions.services.openoffice.org/dictionary`.

To create an English dictionary data set that you can use with the **Text Filter** node or the %TEXTSYN macro, do the following:

1. Go to `http://extensions.services.openoffice.org/en/project/en_US-dict` to download the zipped OpenOffice US English spelling dictionary files.

2. Click the **Get it!** button to download the file en_US.oxt to your local machine. You might need to rename this file to en_US.zip in order to extract the dictionary files. The dictionary file is en_US.dic, and you must extract this file to a location on your local machine.

3. In SAS Enterprise Miner, select **View** ⇨ **Program Editor**.

4. In the Program Editor that opens, enter and run the following SAS code. This code removes extraneous characters from the OpenOffice dictionary, assigns the proper noun part of speech to any term that is capitalized in the file, and creates an English dictionary data set with the required format.

Note: Be sure to change **`<fileLocation>`** to the path to the unzipped dictionary files. Also, note that the following code uses a library called tmlib. You will need to create this library with a LIBNAME statement before running this code, or change tmlib to another library that you have already created.

```
data tmlib.engdict (keep=term pos);
   length inputterm term $32;
   infile '<fileLocation>en_US.dic'
      truncover;
   input linetxt $80.;
   i=1;
   do until (inputterm = ' ');
      inputterm = scan(linetxt, i, ' ');
      if inputterm ne ' ' then do;
          location=index(inputterm,'/');
        if location gt 0 then
        term = substr(inputterm,1,location-1);
        if location eq 0 then
        term = inputterm;
         if lowcase(term) ne term then pos = 'Prop';
         term = lowcase(term);
         output;
         end;
      i=i+1;
      end;
   run;
```

Term Weighting

Contents

Overview of Term Weights

The term weighting options in the **Text Filter** node enable you to vary the importance of terms based on how frequently the terms occur in individual documents and how the terms are distributed throughout the document collection. A weighted, term-by-document frequency matrix is created for a text mining analysis by first assigning child frequencies to their parent terms, then applying a weighting function to the frequency of occurrence of each term in each document, and finally scaling the entry for each term by multiplying it by its term weight. This weighted, term-by-document frequency matrix becomes the underlying representation for the collection. Formally, term and frequency weights are applied as follows.

Let f_{ij} be the ijth frequency in the unweighted term-by-document matrix, the frequencies of terms be weighted using the function g(.), and the weight of the ith term be denoted by w_i. Then the weighted frequency of element f_{ij} in the term-by-document frequency matrix is: $w_i*g(f_{ij})$.

Frequency Weighting Methods

The following frequency weighting functions, g(.), are available in the **Text Filter** node.

- **Default**

  The default frequency weighting method is Log with one exception. In a process flow that has multiple **Text Filter** nodes, the default frequency weighting method that is used in a node is determined by the setting that was specified in the previous **Text Filter** node.

- **Binary**

  an indicator function, where $g(f_{ij})= 1$ if a term appears in the document, and $g(f_{ij})= 0$ if it does not. This function removes the effect of terms that occur repeatedly in the same document.

- **Log**

  $g(f_{ij})= \log_2(f_{ij}+1)$. This function dampens the effect of terms that occur many times in a document.

- **None**

  $g(f_{ij})= 1$. In other words, no change is applied to the raw frequency for the term.

Term Weighting Methods

Term weights are useful for distinguishing important terms from others. In general, the assumption is that terms that are useful for categorizing documents are those that occur in only a few documents but many times in those few documents. SAS Text Miner implements several methods that, at a high level, have the same goals. However, because the functions vary, you should try different weighting techniques to see which method works best for a particular analysis.

In the **Text Filter** node, the following term weights, w_i, are available.

- **Default**

  The default term weighting method is Entropy (if the data source does not have a categorical target) and Mutual Information (if the data source has a categorical target) with one exception. In a process flow that has multiple **Text Filter** nodes, the default term weighting method used in a node is controlled by the setting that was specified in the previous **Text Filter** node.

- **Entropy**

$$w_i = 1 + \sum_j \frac{(f_{ij}/g_i)\cdot\log_2(f_{ij}/g_i)}{\log_2(n)}$$

  Here, g_i is the number of times that term i appears in the document collection, and n is the number of documents in the collection. Log(.) is taken to be 0 if $f_{ij}=0$. This method gives greater weight to terms that occur infrequently in the document collection by using a derivative of the entropy measure found in information theory.

- **Inverse Document Frequency**

  $$w_i = \log_2\left(\frac{1}{P(t_i)}\right) + 1$$

  Here, P(t_i) is the proportion of documents that contain term t_i. This method gives greater weight to terms that occur infrequently in the document collection by placing the number of documents that contain the term in the numerator of the formula.

- **Mutual Information**

  $$w_i = \max_{C_k}\left[\log\left(\frac{P(t_i, C_k)}{P(t_i)\,P(C_k)}\right)\right]$$

  Here, P(t_i) is the proportion of documents that contain term ti, P(C_k) is the proportion of documents that belong to category C_k, and P(t_i ,C_k) is the proportion of documents that contain term ti and belong to category C_k. Log(.) is taken to be 0 if P(t_i ,C_k)=0 or P(C_k)=0.

  This weight is valid only if the data source includes a categorical target variable. The weight is proportional to the similarity of the distribution of documents that contain the term to the distribution of documents that are contained in the respective category.

- **None**

  w_i= 1. In other words, no term weight is applied.

Text Filter Node Search Expressions

The **Text Filter** node and the **Interactive Filter Viewer** enable you to use a search expression to filter documents. A subset of documents is returned that matches your query. A query consists of a list of terms, and the search results show a set of documents that contain at least one of the query terms. A relevance score indicates how well each document matches the query. A relevance of 1 indicates that document was the best match in the collection for that query. You can use the following special characters to enhance a search:

- `+term` returns only documents that include `term`
- `-term` returns only documents that do not include `term`.
- `"text string"` returns only documents that include the quoted text.
- `string1*string2` returns only documents that include a term that begins with `string1`, ends with `string2`, and has text in between.
- `>#term` returns only documents that include `term` or any of the synonyms that have been assigned to the `term`.

Note: The Interactive Filter Viewer does not support use of the **+** operator with the **>#** operator in a search. Query length is limited to 100 bytes on the Solaris for 64 and Solaris for x64 platforms.

Interactive Filter Viewer

Contents

Overview of the Interactive Filter Viewer

The Interactive Filter Viewer enables you to refine the parsed and filtered data that exists after the **Text Filter** node has run. You can save any changes that you make to the data when you close the Interactive Filter Viewer. The edited data is used in subsequent nodes of the analysis. To access the Interactive Filter Viewer, click the ellipsis button that corresponds to the **Filter Viewer** property in the Properties panel of the **Text Filter** node.

Searching in the Interactive Filter Viewer

Search Box Functionality

The Search box enables you to filter documents based on the results of a search expression. If you enter a search expression in the **Search Expression** property before you run the **Text Filter** node, then the Search box is populated with that expression when you open the Interactive Filter Viewer. You can edit the expression or enter a new expression in the Interactive Filter Viewer and click **Apply** to refine the filter. Search expressions can also be added to the Search box by right-clicking on a term in the Terms table and selecting **Add Term to Search Expression**. For more information about search expressions, see "Text Filter Node Search Expressions" on page 59.

Note: The number of search results for a given term might differ from the number of documents assigned to that term in the Terms table because the search is over the original text and not over the entries in the table.

Find Functionality

To access the Find Text window, select the Documents window or the Terms window and select **Edit ⇨ Find**. Enter text that you want to find, and click **OK**, and you will be advanced to the next matching observation in the **TEXT** column of the Documents window, or the **TERM** column of the Terms window. This behavior is different from that of the Search box.

Documents Window of the Interactive Filter Viewer

The Documents window of the Interactive Filter Viewer displays all of the variables in the input data source. The first variable that is displayed is the variable that was used for text parsing. To wrap or unwrap the full contents of this variable in the column cells, select **Edit** ⇨ **Toggle Show Full Text**. You cannot wrap the contents of the other columns.

If the input data source in the process flow diagram contains paths to files that contain the document text (rather than containing the full text, itself) and if these files are accessible from the client, then you can use the **Edit** menu to view the full document text. Select **Edit** ⇨ **View Input File** to view the file stored in the location that is specified by the variable with the role **`Web Address`** (if such a variable exists in the data source). Select **Edit** ⇨ **View Original** to view the file that is stored in the location specified by the variable with the role **`Text Location`**.

Terms Window of the Interactive Filter Viewer

The Terms window of the Interactive Filter Viewer displays a table that is similar to the Terms table displayed in the **Text Filter** node results window. However, fewer variables are displayed. And, instead of displaying only top-level terms, the Terms window of the Interactive Filter Viewer displays all terms. If stemming or synonym lists are used in the node, then parent terms appear as the root of a tree in the table; children appear as branches. Click the plus (+) symbol at the left of a parent term to view or hide its children.

The following functions in the Terms window enable you to refine your filtered data:

- Select two or more terms, and then select **Edit** ⇨ **Treat as Synonyms** to define new synonyms. You are prompted to specify which of the selected terms to treat as the parent in the Create Equivalent Terms window. The other selected terms are treated as children.

- Select one or more child terms, and then select **Edit** ⇨ **Remove Synonyms** to disassociate one or more child terms from a parent.

- Select one or more terms, and then select **Edit** ⇨ **Toggle KEEP** to toggle the keep status of one or more terms. For a single term, you can also accomplish this task by selecting the check box in the **KEEP** column for that term.

Note: You can use the SHIFT and CTRL keys to make multiple selections in the table.

The Terms window is equipped with a quick-find functionality that enables you to scroll quickly to a specific row in a sorted column. To use quick-find, sort the table by a particular column, select any row, and enter the first letter or number of any row to which you want to scroll. The cursor advances to the first item under the sorted column that starts with that letter or number. Quick-find is available for all columns except for **KEEP**. Quick-find complements the find functionality.

Concept Linking Window of the Interactive Filter Viewer

The Concept Linking window enables you to interactively view terms that are associated with a particular term in a hyperbolic tree display. To view the Concept Linking window, select a term in the Terms window and select **Edit** ⇨ **View Concept Links**.

The Concept Linking window is available for only a single term at a time, and for only kept terms (not for dropped terms).

In the Concept Linking window, the selected term appears in the center of the tree, surrounded by the terms that are most highly associated with that term. The width of the lines in the display represents the strength of association between terms; a thicker line indicates a closer association. For more information, see “Strength of Association for Concept Linking” on page 63. Furthermore, you can double-click a term in the display to expand the tree and include terms that are associated with that term.

If you position the mouse pointer over a term, then a tooltip shows two numbers, separated by a forward slash (/). The first number is the number of documents in which the term appears with its immediate predecessor in the display. The second number represents the total number of documents in which the term occurs in the document collection.

A pop-up menu enables you to perform these actions in the Concept Linking window:

- **Copy** — Right-click inside the Concept Linking window and select **Copy** to copy the diagram into the clipboard.
- **Expand Links** — Right-click a node, and select **Expand Links** to expand the links associated with the node.
- **Add Term to Search Expression** — Right-click a node, and select **Add Term to Search Expression** to add the term to the Search box.

Create Synonym Data Sets

Perform the following steps to create a synonym data set that consists of synonym changes that you have made in the Interactive Filter Viewer.

1. Click the ellipsis button for the **Filter Viewer** property of the **Text Filter** node.

   Note: The Interactive Filter Viewer can be opened only after the **Text Filter** node has been successfully run.

2. Select terms in the **Terms** table that you want to treat as synonyms.
3. Right-click the selected terms, and then select **Treat as Synonyms** from the pop-up menu.
4. Select the term in the Create Equivalent Terms dialog box that you want to use as the parent term, and then click **OK**.
5. Select **File** ⇨ **Export Synonyms**.

   The Export Synonyms dialog box appears.

6. Select a library from the **Library** drop-down menu for the data set that you want to create.

   Note: If you do not see a library that you want to store a synonym data set, you can create a library in SAS Enterprise Miner before opening the Interactive Filter Viewer.

7. Provide a name for your synonym data set.

8. Click **OK**.

Strength of Association for Concept Linking

For a given pair of terms, their “strength of association” with one another is computed using the binomial distribution. This strength measure is used to produce concept linking diagrams in the **Interactive Filter Viewer** of the **Text Filter** node.

The following assumptions obtain.

- `n` is the number of documents that contain term `B`
- `k` is the number of documents containing both term `A` and term `B`
- `p = k/n` is the probability that term `A` occurs when term `B` occurs, assuming that they are independent of each other.

Then the strength of association between the terms `A` and `B`, for a given `r` documents, is as follows:

$$Strength = log_e(1/Prob_k)$$

where

$$Prob_k = \sum_{r=k}^{r=n} Prob(r).$$

$$Prob(r) = [n! / [r!(n - r)!]]\ p^r(1-p)^{(n - r)}$$

Chapter 6

The Text Cluster Node

Overview of the Text Cluster Node

The **Text Cluster** node clusters documents into disjoint sets of documents and reports on the descriptive terms for those clusters. Two algorithms are available. The Expectation Maximization algorithm clusters documents with a flat representation, and the Hierarchical clustering algorithm groups clusters into a tree hierarchy. Both approaches rely on the singular value decomposition (SVD) to transform the original weighted, term-document frequency matrix into a dense but low dimensional representation.

The most memory-intensive task of the text clustering process is computing the SVD of the weighted term-by-document frequency matrix. For more information, see "Singular

Value Decomposition” on page 75. When in-memory resources are limited, the node might use, instead of the full collection, a simple random sample of the documents in an attempt to run the node successfully. Sampling occurs only if the node encounters a memory failure during an attempt to compute the SVD without sampling. Furthermore, because sampling generally occurs when the document collection is extremely large, there is typically not an adverse effect on modeling results. Exactly when sampling occurs depends on a number of parameters including the size of your collection, the platform on which your system is running, and the available RAM.

- “Text Cluster Node Input Data” on page 66
- “Text Cluster Node Properties” on page 66
- “Text Cluster Node Results” on page 68
- “Text Cluster Node Output Data” on page 69
- “Using the Text Cluster Node” on page 70

For related topics that you might find useful when using the **Text Cluster** node, see the following:

- “Singular Value Decomposition” on page 75
- “Clustering Techniques” on page 80

Note: The **Text Cluster** node is not supported for use in group processing (Start Groups and End Groups nodes).

Text Cluster Node Input Data

The **Text Cluster** node requires the following predecessor nodes:

- a data source
- a **Text Parsing** node
- a **Text Filter** node

Text Cluster Node Properties

Contents

Text Cluster Node General Properties

These are the general properties that are available on the **Text Cluster** node:

- **Node ID** — displays the ID that is assigned to the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Cluster** node added to a diagram will have the Node ID `TextCluster`, and the second **Text Cluster** node added will have the Node ID `TextCluster2`.

- **Imported Data** — accesses a list of the data sets imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Exported Data** — accesses a list of the data sets exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Cluster Node Train Properties

General Train Properties

These are the training properties that are available on the **Text Cluster** node:

- **Variables** — accesses a list of variables and associated properties in the data source. Click the ellipsis button to open the Variables window.
- "Transform Properties" on page 67
- "Cluster Properties" on page 67

Transform Properties

- **SVD Resolution** — specifies the resolution to use to generate the singular-value decomposition (SVD) dimensions. For more information about SVD, see "Singular Value Decomposition" on page 75.
- **Max SVD Dimensions** — specifies the maximum number of SVD dimensions to generate. The minimum value that you can specify is `2`, and the maximum value that you can specify is `500`.

Cluster Properties

- **Exact or Maximum Number** — specifies whether to find an exact number of clusters or any number less than or equal to a maximum number of clusters.

- **Number of Clusters** — specifies the number of clusters to create; this is the exact number if the value of **Exact or Maximum Number** is `Exact`, and it is the maximum number if the value of **Exact or Maximum Number** is `Maximum`.
- **Cluster Algorithm** — specifies the clustering algorithm to use. For more information about clustering, see "Clustering Techniques" on page 80.
- **Descriptive Terms** — specifies the number of descriptive terms to display for each cluster. The default value is `15`. For more information, see "Descriptive Terms" on page 82.

Text Cluster Node Status Properties

These are the status properties that are displayed on the **Text Cluster** node:

- **Create Time** — time that the node was created.
- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.
- **Last Error** — error message, if any, from the last run.
- **Last Status** — last reported status of the node.
- **Last Run Time** — time at which the node was last run.
- **Run Duration** — length of time required to complete the last node run.
- **Grid Host** — grid host, if any, that was used for computation.
- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner extension node. The value of this property is always `No` for the **Text Cluster** node.

Text Cluster Node Results

Contents

Results Window for the Text Cluster Node

After the **Text Cluster** node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status dialog box that opens immediately after a successful run of the **Text Cluster** node.

- Click the Results icon on the main Toolbar.
- Right-click the **Text Cluster** node, and select **Results**.

The Results window for the **Text Cluster** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are the same as other nodes. However, the **View** menu does not include the selections **Assessment** or **Model**. Instead, it includes **Clusters**, which access submenus that list the graphical and tabular results.

Note: You can access the SAS log from the **SAS Results** submenu of the **View** menu. This log can be a useful debugging tool.

Text Cluster Node Graphical and Tabular Results

The following are the graphical results in the **Text Cluster** node Results window.

- The **Cluster Frequencies** pie chart shows the frequency of each cluster. Position the mouse pointer over a sector to see the cluster ID and frequency in a tooltip. Select a sector to highlight the cluster in the other graphical results windows and the **Clusters** table.
- The **Cluster Frequency by RMS** scatter plot shows the root mean squared (RMS) standard deviation by frequency. Position the mouse pointer over a point to see the frequency, RMS standard deviation, and descriptive terms in the cluster in a tooltip. Select a point to highlight the cluster in the other graphical results windows and the **Clusters** table.
- The **Distance Between Clusters** scatter plot shows the distance between clusters using a Cartesian coordinate system. Position the mouse pointer over a point to see the x-coordinate, y-coordinate, cluster ID, and descriptive terms in the cluster in a tooltip. Select a point to highlight the cluster in the other graphical results windows and the **Clusters** table.
- The **Cluster Hierarchy** plot shows a hierarchical relationship among clusters when the **Cluster Algorithm** property is set to **`HIERARCHICAL`**. Select a point to highlight corresponding information in the **Hierarchy Data** table.

The **Clusters** table displays information about each cluster.

The **Hierarchy Data** table displays hierarchical and statistical information about each cluster in the Clusters table when the **Cluster Algorithm** property is set to **`HIERARCHICAL`**.

Text Cluster Node SAS Output Results

The SAS output from the **Text Cluster** node includes summary information about the input variables.

Text Cluster Node Output Data

For information about output data for the **Text Cluster** node, see "Output Data for SAS Text Miner Nodes" on page 127.

Using the Text Cluster Node

This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

This example uses the **Text Cluster** node to cluster SAS Users Group International (SUGI) abstracts.

Note: SAS Users Group International is now SAS Global Forum.

Perform the following steps:

1. Create a data source for SAMPSIO.ABSTRACT. Change the Role of the variable TITLE to **`ID`**.

   Note: The SAMPSIO.ABSTRACT data set contains information about 1,238 papers prepared for meetings of SUGI from 1998 through 2001 (SUGI 23 through 26). The variable TITLE is the title of the SUGI paper. The variable TEXT contains the abstract of the SUGI paper.

2. Add the SAMPSIO.ABSTRACT data source to the diagram workspace.

3. Select the **Text Mining** tab on the Toolbar, and drag a **Text Parsing** node into the diagram workspace.

4. Connect the Input Data node to the **Text Parsing** node.

5. Select the **Text Parsing** node, and then click the ellipsis for the **Stop List** property.

6. Click the **Import** button, browse to select SAMPSIO.SUGISTOP as the stop list, and then click **OK**. Click **OK** to exit the dialog box for the **Stop List** property.

7. Set the **Find Entities** property to **Standard**.

8. Click the ellipsis button for the **Ignore Types of Entities** property to open the Ignore Types of Entities dialog box.

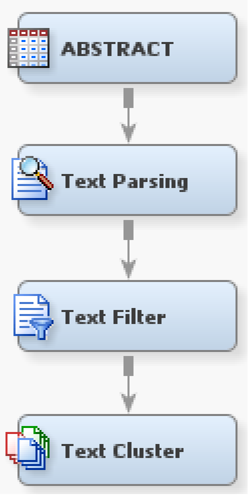
9. Select all entity types except for: **`Location`**, **`Organization`**, **`Person`**, and **`Product`**. Click **OK**.

10. Select the **Text Mining** tab, and drag a **Text Filter** node into the diagram workspace.

11. Connect the **Text Parsing** node to the **Text Filter** node.

12. Select the **Text Mining** tab, and drag a **Text Cluster** node into the diagram workspace.

13. Connect the **Text Filter** node to the **Text Cluster** node. Your process flow diagram should resemble the following:

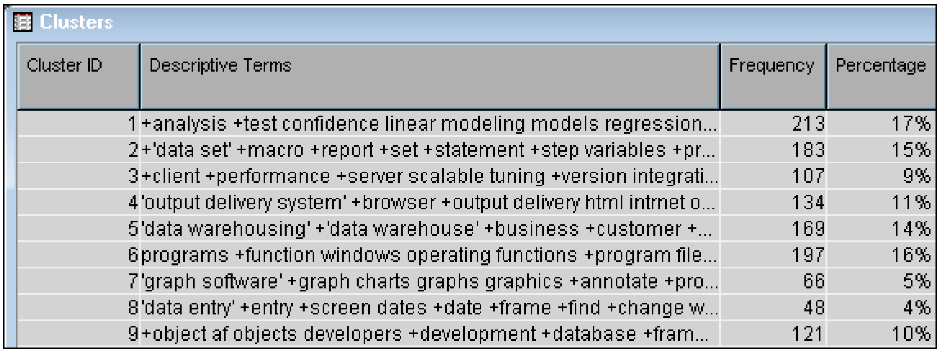


14. Right-click the **Text Cluster** node and select **Run**. Click **Yes** in the Confirmation dialog box.

15. Click **Results** in the Run Status dialog box when the node has finished running.

16. Select the Clusters table.

    The Clusters table contains an ID for each cluster, the descriptive terms that make up that cluster, and statistics for each cluster.

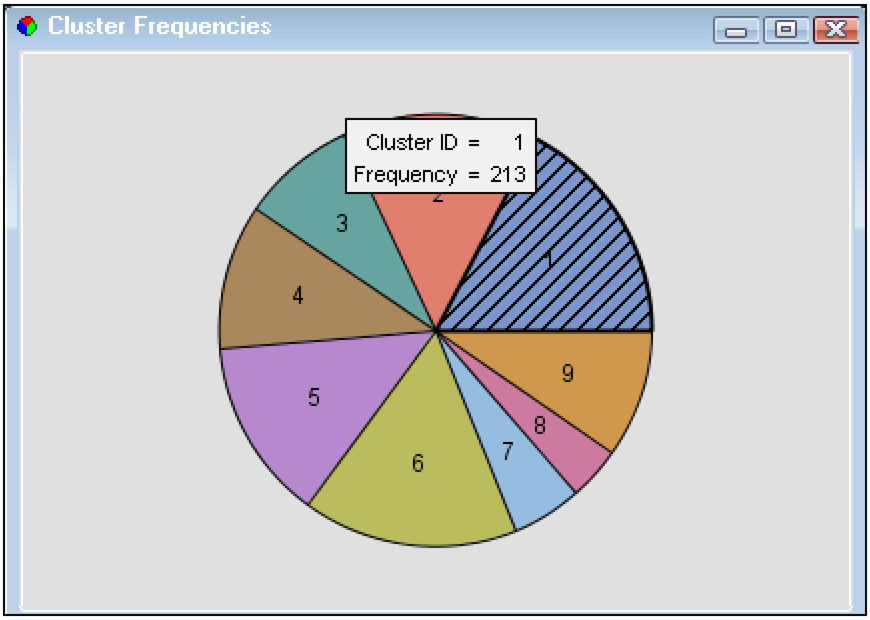


Clusters

Cluster ID	Descriptive Terms	Frequency	Percentage
1	+analysis +test confidence linear modeling models regression...	213	17%
2	+'data set' +macro +report +set +statement +step variables +pr...	183	15%
3	+client +performance +server scalable tuning +version integrati...	107	9%
4	'output delivery system' +browser +output delivery html intrnet o...	134	11%
5	'data warehousing' +'data warehouse' +business +customer +...	169	14%
6	programs +function windows operating functions +program file...	197	16%
7	'graph software' +graph charts graphs graphics +annotate +pro...	66	5%
8	'data entry' +entry +screen dates +date +frame +find +change w...	48	4%
9	+object af objects developers +development +database +fram...	121	10%

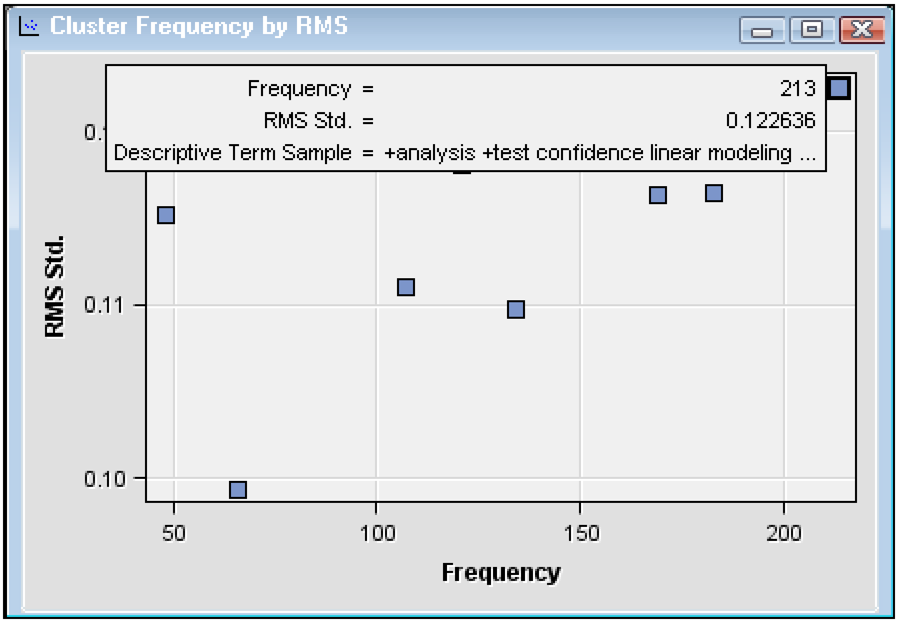
17. Select the first cluster in the Clusters table.

18. Select the Cluster Frequencies window to see a pie chart of the clusters by frequency. Position the mouse pointer over a section to see the frequency for that cluster in a tooltip.



Cluster ID = 1
Frequency = 213

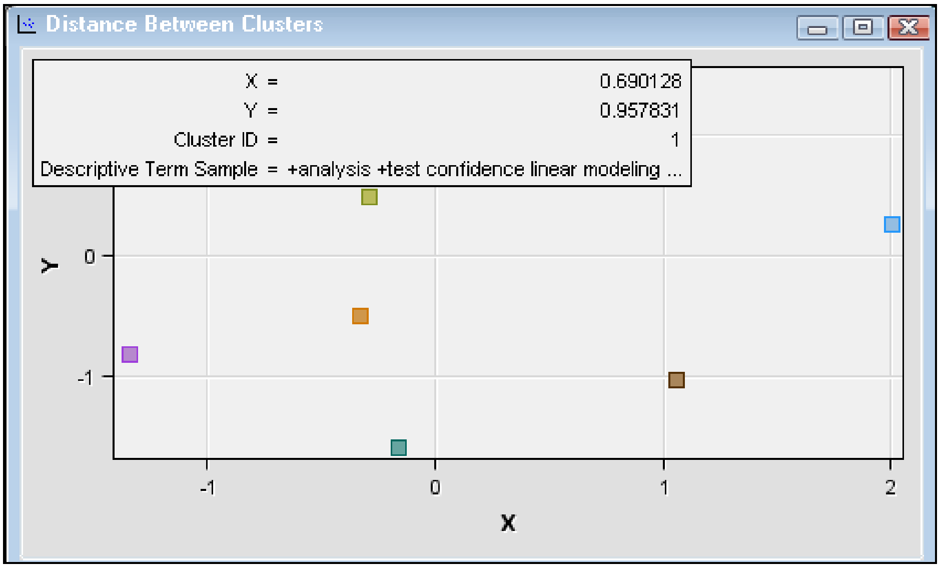
19. Select the Cluster Frequency by RMS window, and then position the mouse pointer over the highlighted cluster.



Frequency = 213
RMS Std. = 0.122636
Descriptive Term Sample = +analysis +test confidence linear modeling ...

While the frequency of the first cluster is the highest, how does it compare to the other clusters in terms of distance?

20. Select the Distance Between Clusters window, and then position the mouse pointer over the highlighted cluster to see the position of the first cluster in an X and Y coordinate grid.



Position the mouse pointer over other clusters to compare distances.

21. Close the Results window.

Now compare the clustering results obtained with the Expectation-Maximization clustering algorithm with using a Hierarchical clustering algorithm.

22. Select the **Text Cluster** node.

23. Select `Exact` for the **Exact or Maximum Number** property.

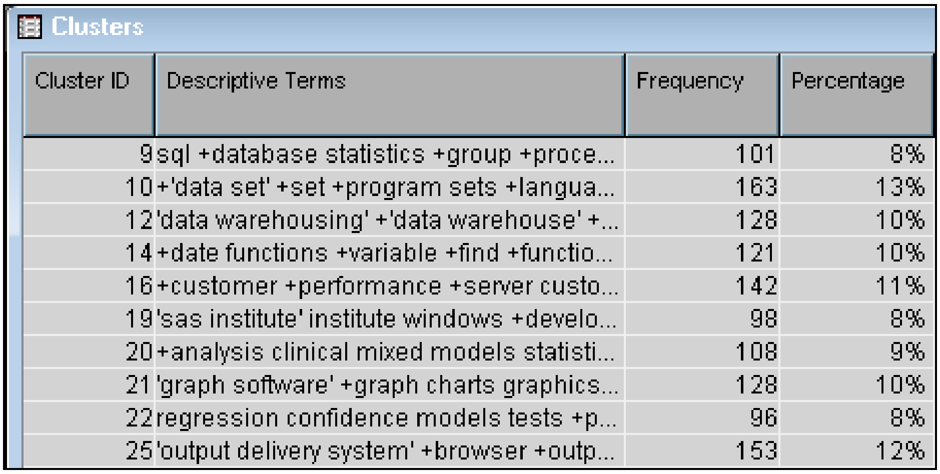
24. Specify *10* for the **Number of Clusters** property.

25. Select `Hierarchical` for the **Cluster Algorithm** property.

26. Right-click the **Text Cluster** node and select **Run**. Click **Yes** in the Confirmation dialog box.

27. Click **Results** in the Run Status dialog box when the node has finished running.

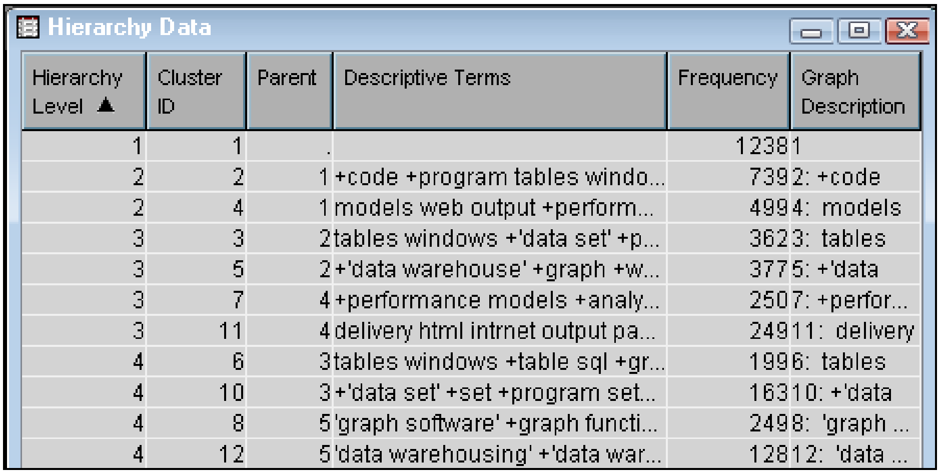
28. Select the Clusters table.



Cluster ID	Descriptive Terms	Frequency	Percentage
9	sql +database statistics +group +proce...	101	8%
10	+'data set' +set +program sets +langua...	163	13%
12	'data warehousing' +'data warehouse' +...	128	10%
14	+date functions +variable +find +functio...	121	10%
16	+customer +performance +server custo...	142	11%
19	'sas institute' institute windows +develo...	98	8%
20	+analysis clinical mixed models statisti...	108	9%
21	'graph software' +graph charts graphics...	128	10%
22	regression confidence models tests +p...	96	8%
25	'output delivery system' +browser +outp...	153	12%

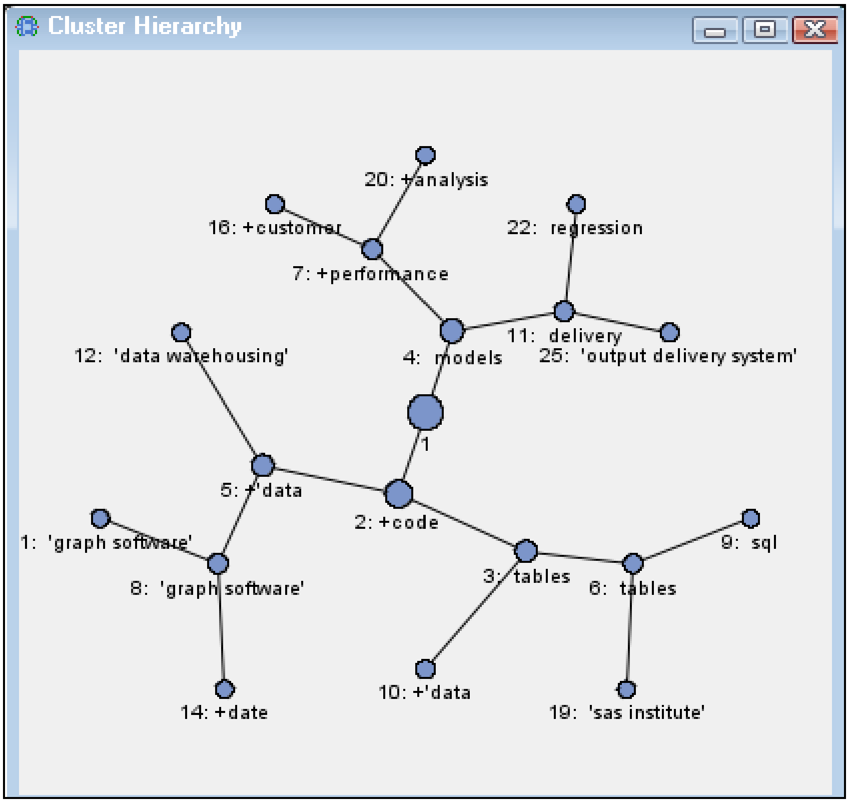
Notice that while there are 10 clusters in the table, the Cluster IDs do not range from 1 to 10.

29. Select the Hierarchy Data table for more information about the clusters that appear in the Clusters table.


Hierarchy Data

Hierarchy Level ▲	Cluster ID	Parent	Descriptive Terms	Frequency	Graph Description
1	1	.		1238	1
2	2	1	+code +program tables windo...	739	2: +code
2	4	1	models web output +perform...	499	4: models
3	3	2	tables windows +'data set' +p...	362	3: tables
3	5	2	+'data warehouse' +graph +w...	377	5: +'data
3	7	4	+performance models +analy...	250	7: +perfor...
3	11	4	delivery html intrnet output pa...	249	11: delivery
4	6	3	tables windows +table sql +gr...	199	6: tables
4	10	3	+'data set' +set +program set...	163	10: +'data
4	8	5	'graph software' +graph functi...	249	8: 'graph ...
4	12	5	'data warehousing' +'data war...	128	12: 'data ...

30. Select the Cluster Hierarchy table for a hierarchical graphical representation of the clusters.



31. Close the Results window.

Singular Value Decomposition

Parsing the document collection generates a term-document frequency matrix. Each entry of the matrix represents the number of times that a term appears in a document. For a collection of several thousand documents, the term-document frequency matrix can contain hundreds of thousands of words. It requires too much computing time and space to analyze this matrix effectively. Also, dealing with high dimensional data is inherently difficult for modeling. To improve the performance, singular value decomposition (SVD) can be implemented to reduce the dimensions of the term-document frequency matrix by transforming the matrix into a lower dimensional, more compact, and informative form.

A high number of SVD dimensions usually summarize the data better, but the higher the number, the more computing resources are required. The **Text Cluster** node determines the number of SVD dimensions based on the values of the **SVD Resolution** and **Max SVD Dimensions** properties. The value of the **SVD Resolution** property can be set to **`Low`** (default), **`Medium`**, or **`High`**. High resolution yields more SVD dimensions. The default value of the **Max SVD Dimensions** property is 100, and the value must be between 2 and 250. Suppose the maximum number of SVD dimensions that you specify for the **Max SVD Dimensions** property is maxdim and these maxdim SVD dimensions account for p% of the total variance. High resolution always generates the maximum number of SVD dimensions, maxdim. For medium resolution, the recommended number of SVD dimensions account for 5/6*(p% of the total variance). For low resolution, the recommended number of SVD dimensions account for 2/3*(p% of the total variance).

The computation of the SVD is itself a memory-intensive task. For extremely large problems the SVD might automatically perform a random sample of the documents in an attempt to avoid running out of memory. When this occurs, a note indicating that sampling has occurred will be written to the SAS log.

The SVD approximates the original weighted frequency matrix. It is the best least squares fit to that matrix. In other words, for any given k, the transformation output will be the factorization of the matrix with k dimensions that best approximates the original matrix. A higher value of k gives a better approximation to the matrix A. 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 200 are appropriate unless the document collection is small. In SAS Text Miner, you can specify the number of dimensions (k). That is, you can specify the first k singular values to be calculated. The algorithm for computing the singular values is designed to give only the leading singular values. The value for k can be at most 4 fewer than the minimum of the number of rows and number of columns of A. In some cases, the algorithm might not be able to calculate that many singular values, so you must reduce the number of dimensions. As you carry out text mining, this problem does not usually occur. For your specific text mining application, you might want to compare the results for several values of k. 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.

SVD factors the large, sparse term-by-document frequency matrix by calculating a truncated SVD of the matrix. Then, it projects the rows or columns of the sparse matrix onto the columns of a dense matrix.

Suppose **`A`** is the large, sparse term-by-document frequency matrix with weighted entries. The SVD of a matrix **`A`** is a factorization of **`A`** into three new matrices **`U`**, **`D`**, and **`V`** such that $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$, where matrices U and V have orthonormal columns, and D is a diagonal matrix of singular values. SVD calculates only the first k columns of these matrices (U, D, and V). This is called the truncated decomposition of the original matrix.

After the SVD is computed, each column (or document) in the term-by-document frequency matrix can be projected onto the first k columns of U. Mathematically, this projection forms a k-dimensional subspace that is a best fit to describe the data set. Column projection (document projection) of the term-by-document matrix is a method to represent each document by k distinct concepts.

In other words, the collection of documents is mapped into a k-dimensional space in which one dimension is reserved for each concept. Similarly, each row (or term) in the term-by-document matrix can be projected onto the first k columns of V.

The following description shows the benefits of the SVD. Suppose we 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. A 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 document 4 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 — river boat floats up the river
- Document 5 — boat is by the dock near the bank
- Document 6 — with credit, I can borrow cash from the bank
- Document 7 — boat floats by dock near the river bank
- Document 8 — check the parade route to see the floats
- Document 9 — along the parade route

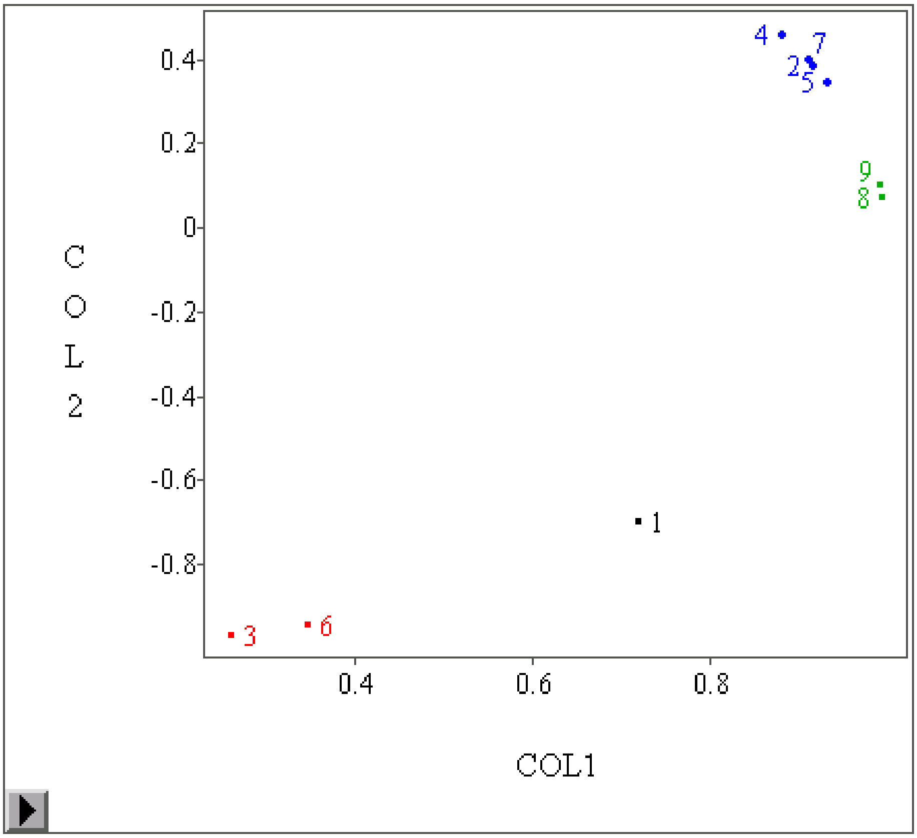
Parsing this document collection generates the following term-by-document frequency matrix:

	d1	**d2**	**d3**	**d4**	**d5**	**d6**	**d7**	**d8**	**d9**
the	2	2	0	1	2	1	1	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

	d1	d2	d3	d4	d5	d6	d7	d8	d9
on	0	1	1	0	0	0	0	0	0
borro w	0	0	1	0	0	1	0	0	0
credit	0	0	1	0	0	1	0	0	0
+ floats	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

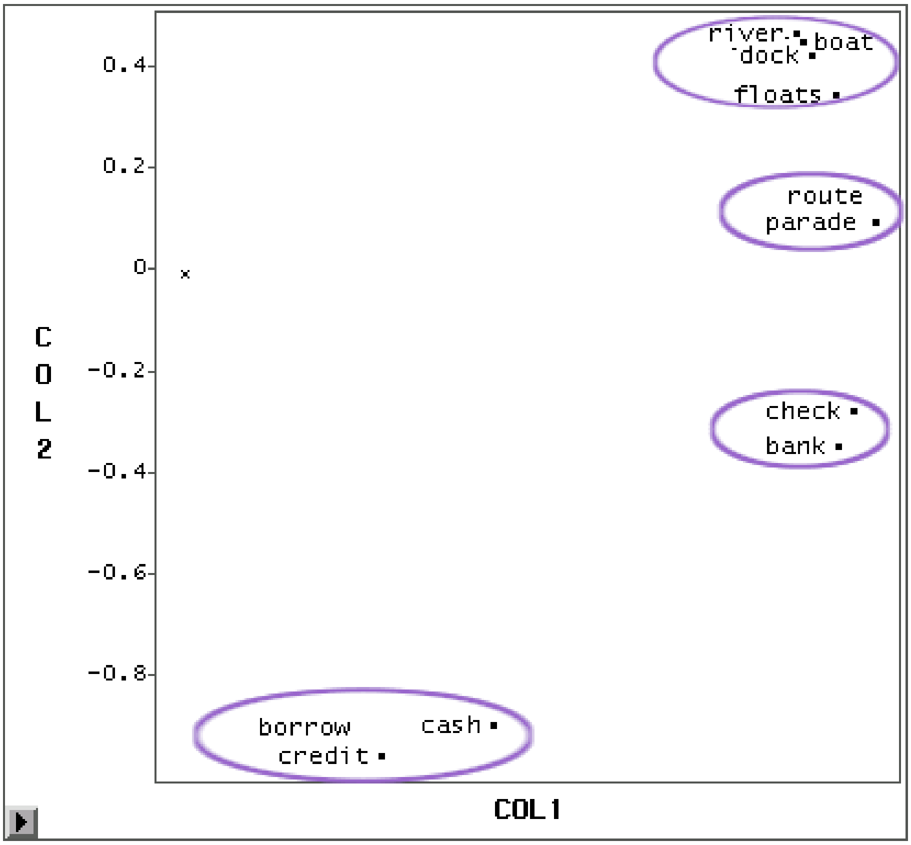
By using the co-occurrence of items from the matrix as a measure of similarity, you can see that documents 1 and 2 are more similar than documents 1 and 3. This is because documents 1 and 2 share the same word **`bank`**, but documents 1 and 3 have no words in common. However, in fact, documents 1 and 2 are not related at all, but documents 1 and 3 are similar. The SVD helps overcome this difficulty.

The SVD is then applied to approximate the above matrix, and documents are 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 displays show the two-dimensional scatter plot of documents.

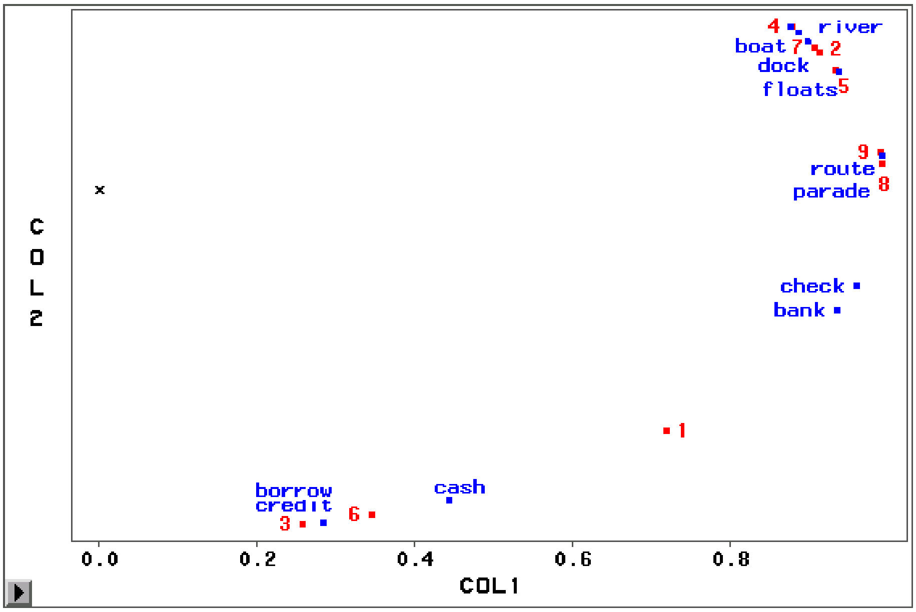


Document 1 is closer to document 3 than it is to document 2. This is true even though documents 1 and 3 do not share any of the same words. On the other hand, document 5 is directly related to documents 2, 4, and 7. That is, projections tend to place similar documents — even if they share few common words — close to one another in the reduced space. The SVD represents terms with 2 dimensions rather than the original 16 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:



Clustering Techniques

Contents

Hierarchical Clustering

When the **Text Cluster** node performs hierarchical clustering on document vectors, it obtains an initial set of seeds. Then, the node generates a result tree. In this tree, parent clusters always contain at least two children, and children are either leaves or have subtrees, themselves.

In hierarchical clustering, the Ward minimum variance method is used. In this method, the distance between two clusters is as follows:

$$\frac{(u_1 - u_2)'(u_1 - u_2)}{1/n_1 + 1/n_2}$$

Here, u_1 and u_2 are the cluster means, and n_1 and n_2 are the cluster frequencies.

Expectation-Maximization Clustering

The expectation-maximization (EM) algorithm assumes that a mixture model approximates the data distribution by fitting k cluster density functions, f_h (h=1, ..., k), to a data set with d variables. The mixture model probability density function evaluated at point x is as follows:

$$p(x) = \sum_{h=1}^{k} w_h f_h(x \mid \mu_h, \Sigma_h)$$

Here, w_h is the proportion of data that belongs to primary cluster h. Each cluster is modeled by a d-dimensional Gaussian probability distribution as follows:

$$f_h(x \mid \mu_h, \Sigma_h) = \frac{1}{\sqrt{(2\pi)^d \left|\Sigma_h\right|}} \exp\left(-\frac{1}{2}(x - \mu_h)^T (\Sigma_h)^{-1} (x - \mu_h)\right)$$

Here, μ_h and D_h are the mean vector and covariance matrix for each cluster h.

In the **Text Cluster** node, EM clustering is an iterative process:

1. Obtain initial parameter estimates.

2. Apply the standard or scaled version of the EM algorithm to find primary clusters and to update parameter estimates. The standard EM algorithm uses the entire input data set at each iteration to estimate the model parameters. The scaled EM algorithm, on the other hand, uses a portion of the input data set at each iteration.

   If the data set is small enough and less memory is needed, the standard EM algorithm is used. Otherwise, the scaled EM algorithm is used. The algorithm terminates when two successive log-likelihood values differ by a particular amount or when a maximum of five iterations has been reached.

3. (For the scaled EM algorithm) Summarize data in the primary summarization phase. Observations that are near each of the primary cluster means are summarized. If a primary cluster contains fewer observations than the minimum requirement, this cluster becomes inactive. Inactive clusters are not used for updating the parameter estimates in the EM algorithm.

4. (For the scaled EM algorithm) Summarize data in the secondary summarization phase. Secondary clusters are identified as subsets of observations that result from a k-means clustering algorithm, where the sample standard deviation for each variable does not exceed the specified amount. Then, a hierarchical agglomerative clustering algorithm is used to combine similar secondary clusters. A secondary cluster is disjoint from all other secondary clusters and from all primary clusters.

For each observation x in the data set at iteration j, the parameter estimates of a standard EM algorithm are computed as follows:

1. Compute the membership probability of x in each cluster h = 1,...,k.

$$w_h^j(x) = \frac{w_h^j f_h\left(x \mid \mu_h^j, \Sigma_h^j\right)}{\sum_i w_i^j f_i\left(x \mid \mu_i^j, \Sigma_i^j\right)}$$

2. Update the mixture model parameters for each cluster h = 1,...,k.

$$w_h^{j+1} = \sum_x w_h^j(x)$$

$$\mu_h^{j+1} = \frac{\sum\limits_x w_h^j(x)x}{\sum\limits_x w_h^j(x)} \quad \Sigma_h^{j+1} = \frac{\sum\limits_x w_h^j(x)(x-\mu_h^{j+1})(x-\mu_h^{j+1})^T}{\sum\limits_x w_h^j(x)}$$

The iterative computation stops if $\left| L(\Phi^j) - L(\Phi^{j+1}) \le \varepsilon \right|$, where ε > 0, and

$$L(\Phi) = \sum_x \log\left[\sum_{h=1}^{k} w_h f_h(x \mid \mu_h, \Sigma_h)\right]$$

For EM clustering, the distance between a document and a cluster is the Mahalanobis distance, sqrt((x-u)'S(x-u)). Here, u is the cluster mean and S is the inverse of the cluster covariance matrix.

Descriptive Terms

The **Text Cluster** node uses a descriptive terms algorithm to describe the contents of both EM clusters and hierarchical clusters. If you specify to display m descriptive terms for each cluster, then the top 2*m most frequently occurring terms in each cluster are used to compute the descriptive terms.

For each of the 2*m terms, a binomial probability for each cluster is computed. The probability of assigning a term to cluster j is prob=F(k|N, p). Here, F is the binomial cumulative distribution function, k is the number of times that the term appears in cluster j, N is the number of documents in cluster j, p is equal to (sum-k)/(total-N), sum is the total number of times that the term appears in all the clusters, and total is the total number of documents. The m descriptive terms are those that have the highest binomial probabilities.

Descriptive terms must have a keep status of Y and must occur at least twice (by default) in a cluster. You can use the macro variable TM_MINDESCTERMS to change the minimum frequency that a descriptive term must have. If a cluster consists of blank documents or documents that contain only the dropped terms, or if the terms do not meet the minimum required frequency, then no descriptive term will be displayed. For more information about macro variables, see “Using Macro Variables to Set Additional Properties” on page 129.

Chapter 7

The Text Topic Node

Overview of the Text Topic Node

The **Text Topic** node enables you to explore the document collection by automatically associating terms and documents according to both discovered and user-defined topics. Topics are collections of terms that describe and characterize a main theme or idea. The approach is different from clustering because clustering assigns each document to a unique group while the **Text Topic** node assigns a score for each document and term to each topic. Then thresholds are used to determine whether the association is strong enough to consider that the document or term belongs to the topic. As a result, documents and terms can belong to more than one topic or to none at all. The number of topics that you request should be directly related to the size of the document collection (for example, a large number for a large collection).

The most memory-intensive task is computing the singular value decomposition (SVD) of the term-by-document frequency matrix. For more information, see "Singular Value Decomposition" on page 75. When in-memory resources are limited, the **Text Topic** node might use, instead of the full collection, a simple random sample of the documents in an attempt to run the node successfully. Sampling occurs only if the node encounters a memory failure during an attempt to compute the SVD without sampling. Furthermore, because sampling generally occurs when the document collection is extremely large, there is typically not an adverse effect on modeling results. Exactly when sampling occurs depends on a number of parameters including the size of your collection, the platform on which your system is running, and the available RAM.

See the following for more information about the **Text Topic** node.

- "Text Topic Node Input Data" on page 84
- "Text Topic Node Properties" on page 84
- "Text Topic Node Results" on page 87
- "Text Topic Node Output Data" on page 88
- "Using the Text Topic Node" on page 88

For related topics that you might find useful when using the **Text Topic** node, see the following:

- "User-Defined Topic Lists" on page 94
- "Interactive Topic Viewer" on page 95

Note: The **Text Topic** node is not supported for use in group processing (Start Groups and End Groups nodes).

Text Topic Node Input Data

The **Text Topic** node must be preceded by a **Text Parsing** node. If it is not also preceded by a **Text Filter** node, then the **Text Topic** node weights terms using the Log frequency weighting and a term weighting of Mutual Information if there is a categorical target, or Entropy otherwise. For more information, see "Term Weighting" on page 57.

Text Topic Node Properties

Contents

- "Text Topic Node General Properties" on page 85
- "Text Topic Node Train Properties" on page 85
- "Text Topic Node Status Properties" on page 86

Text Topic Node General Properties

These are the general properties that are available on the **Text Topic** node:

- **Node ID** — displays the ID that is assigned to the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Topic** node added to a diagram will have the Node ID `TextTopic`, and the second **Text Topic** node added will have the Node ID `TextTopic2`.
- **Imported Data** — accesses a list of the data sets imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Exported Data** — accesses a list of the data sets exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Topic Node Train Properties

General Train Properties

These are the training properties that are available on the **Text Topic** node:

- **Variables** — accesses a list of variables and associated properties in the data source. Here you can select the text variable and target variable you want to use for Mutual Information weighting. Click the ellipsis button to open the Variables window.
- **User Topics** — specifies a SAS data set that contains user-defined topics. For more information, see "User-Defined Topic Lists" on page 94. A sample data set is provided for use with the SAMPSIO.abstract sample input data set. For more information, see "SAS Text Miner Sample Data Sets" on page 9. Click the ellipsis button to open a window in which you can do the following:
  - import a user-defined topic data set
  - (if a user-defined topic data set is selected) add, delete, and edit user-defined topics

  Two sample data sets are provided for your use. One is for use with the SAMPSIO.ABSTRACT sample input data set, and the other is for detecting positive and negative tone in any English document set. For more information, see "SAS Text Miner Sample Data Sets" on page 9.

Term Topics Properties

- **Number of Single-term Topics** — specifies the maximum number of single-term topics to create from top-weighted terms. This number should be less than or equal to the smaller of 1000 or the number of terms that are imported into the **Text Topic** node. When the node is run, the number of single-term topics actually created is equal to the number specified here plus the number of user topics, u. Then the u single-term topics that are most closely related to user topics are eliminated.

Learned Topics Properties

- **Number of Multi-term Topics** — specifies the maximum number of multi-term topics to create from a rotated singular-value decomposition (SVD) of the weighted term-by-document matrix. For more information, see "Singular Value Decomposition" on page 75. This number should be less than or equal to the smaller of 1000, the number of documents less 6, and the number of terms that are imported into the Text Topic node less 6. When the node is run, the number of multi-term topics actually created is equal to the number specified here plus the number of user topics, u. Then the u multi-term topics that are most closely related to user topics are eliminated. See the *Getting Started with SAS Text Miner* for how that makes it possible to discover new topics that come up at new time periods.

  Note: Convergence problems can arise if you specify a number that is too close to the smaller of the number of documents and the number of terms that are imported into the **Text Topic** node. In this case, choosing a smaller number of multi-term topics can lead to convergence.

- **Correlated Topics** — specifies whether learned topics must be orthogonal (uncorrelated) or if they can be correlated. The topics can align more closely with the individual terms if correlated topics is set to `No`, but then the results should not be fed into a Memory Based Reasoning (MBR) modeling tool that requires orthogonal inputs.

Results Properties

- **Topic Viewer** — (after the node has run) opens the Interactive Topic Viewer in which you can interactively adjust the results of the topic creation. Click the ellipsis button to open the Interactive Topic Viewer. For more information, see "Interactive Topic Viewer" on page 95.

Text Topic Node Status Properties

These are the status properties that are displayed on the **Text Topic** node:

- **Create Time** — time that the node was created.
- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.
- **Last Error** — error message, if any, from the last run.
- **Last Status** — last reported status of the node.
- **Last Run Time** — time at which the node was last run.
- **Run Duration** — length of time required to complete the last node run.
- **Grid Host** — grid host, if any, that was used for computation.
- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner Extension node. The value of this property is always `No` for the **Text Topic** node.

Text Topic Node Results

Contents

Results Window for the Text Topic Node

After the Text Topic node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status window that opens immediately after a successful run of the **Text Topic** node.
- Click the **Results** icon on the main toolbar.
- Right-click the **Text Topic** node, and select **Results**.

The Results window for the **Text Topic** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are the same as other nodes. However, the **View** menu for the **Text Topic** node Results window does not include the selections **Assessment** or **Model**. Instead, it includes **Custom Reports**, which accesses a submenu that lists the graphical and tabular results.

Note: You can access the SAS log that was generated by the node processing from the SAS Results submenu of the View menu. This log can be a useful debugging tool.

Text Topic Node Graphical and Tabular Results

The following are the graphical results in the Text Topic node Results window:

- The **Number of Documents by Topics** bar chart displays the total number of documents in the document collection, broken down by topic. Each bar represents a topic. If you position the mouse pointer over a bar, then a tooltip indicates the topic, the number of documents included in that topic, the category, and the topic ID.
- The **Number of Terms by Topics** bar chart displays the total number of terms in the document collection, broken down by topic. Each bar represents a topic. If you position the mouse pointer over a bar, then a tooltip indicates topic, the number of terms included in that topic, the category, and the topic ID.

The tabular result in the **Text Topic** node Results window is the **Topics** table, which displays information about identified topics. All graphical results in the **Text Topic** node Results window are linked to this table. Therefore, you can select observations in the Topics table and the associated data points are highlighted in the graphics. Or, you can select data points in the graphics and the associated observations are highlighted in the Topics table.

Table 7.1 Contents of the Topics Table

Variable	Description
Topic ID	key value of the topic.
Document Cutoff	minimum topic membership that a document must have to be included in this topic.
Term Cutoff	minimum topic weight that a term must have to be used as a term for this topic. The weight of any term that has an absolute value weight less than this is effectively set to zero.
Topic	terms that describe the topic.
Number of Terms	number of terms in the topic
# Docs	number of documents that contain the topic
Category	user if the topic is user-defined, single if the topic is a single-term topic, and multiple if the topic is a multi-term topic.

Text Topic Node SAS Output Results

The SAS output from the **Text Topic** node includes summary information about the input variables.

Text Topic Node Output Data

For information about output data for the **Text Topic** node, see "Output Data for SAS Text Miner Nodes" on page 127.

Using the Text Topic Node

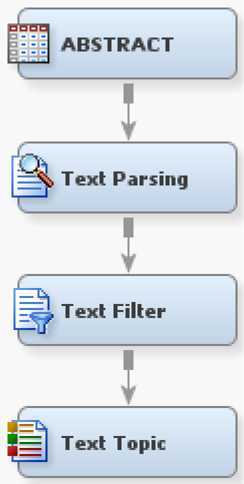
This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

The **Text Topic** node enables you to create topics of interest from a list of terms. The goal in creating a list of topics is to establish combinations of words that you are interested in analyzing. For example, you might be interested in mining articles that discuss the activities of a "company president." One way to approach this task is to look at all articles that have the term "company," and all articles that have the term "president." The **Text Topic** node enables you to combine the terms "company" and "president" into the topic "company president.”

The ability to combine individual terms into topics can improve your text mining analysis . Through combining, you can narrow the amount of text that is subject to analysis to specific groupings of words that you are interested in. This example shows you how to create topics using the **Text Topic** node. This example assumes that you have performed “Using the Text Filter Node” on page 51, and builds off the process flow diagram created there.

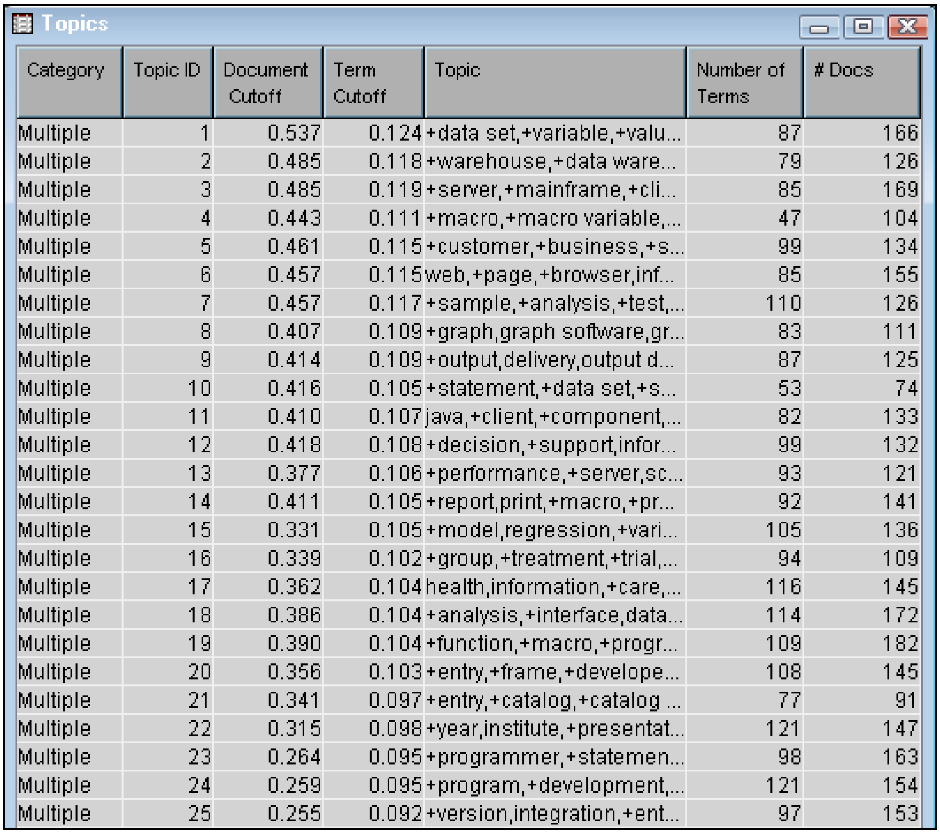
1. Select the **Text Mining** tab on the toolbar, and drag a **Text Topic** node into the diagram workspace.

2. Connect the **Text Filter** node to the **Text Topic** node.



3. In the diagram workspace, right-click the **Text Topic** node and select **Run**. Click **Yes** in the Confirmation dialog box that appears. Click **Results** in the Run Status dialog box when the node finishes running.

4. Select the Topics table to view the topics that have been created with a default run of the **Text Topic** node.

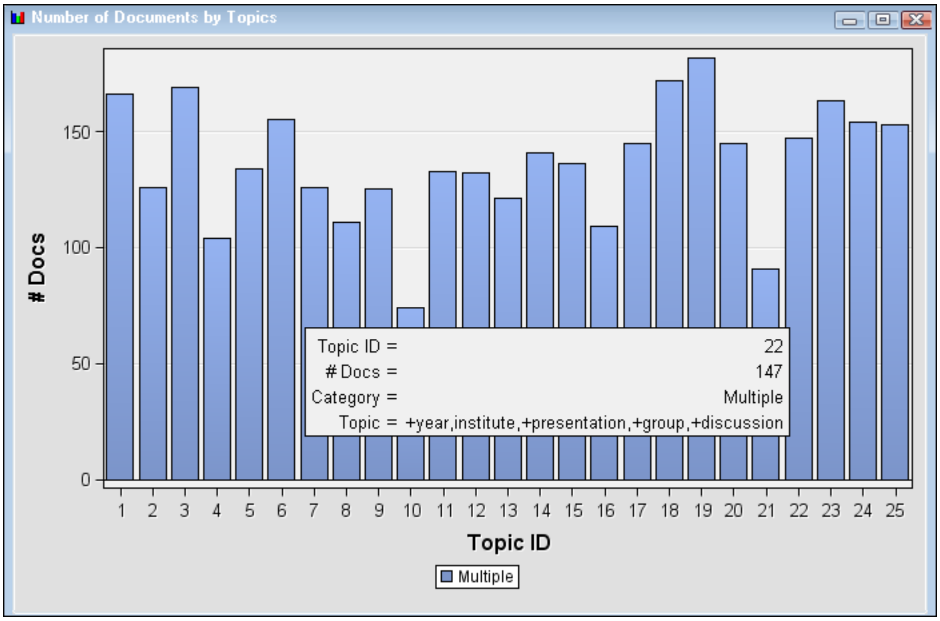


Topics

Category	Topic ID	Document Cutoff	Term Cutoff	Topic	Number of Terms	# Docs
Multiple	1	0.537	0.124	+data set,+variable,+valu...	87	166
Multiple	2	0.485	0.118	+warehouse,+data ware...	79	126
Multiple	3	0.485	0.119	+server,+mainframe,+cli...	85	169
Multiple	4	0.443	0.111	+macro,+macro variable,...	47	104
Multiple	5	0.461	0.115	+customer,+business,+s...	99	134
Multiple	6	0.457	0.115	web,+page,+browser,inf...	85	155
Multiple	7	0.457	0.117	+sample,+analysis,+test,...	110	126
Multiple	8	0.407	0.109	+graph,graph software,gr...	83	111
Multiple	9	0.414	0.109	+output,delivery,output d...	87	125
Multiple	10	0.416	0.105	+statement,+data set,+s...	53	74
Multiple	11	0.410	0.107	java,+client,+component,...	82	133
Multiple	12	0.418	0.108	+decision,+support,infor...	99	132
Multiple	13	0.377	0.106	+performance,+server,sc...	93	121
Multiple	14	0.411	0.105	+report,print,+macro,+pr...	92	141
Multiple	15	0.331	0.105	+model,regression,+vari...	105	136
Multiple	16	0.339	0.102	+group,+treatment,+trial,...	94	109
Multiple	17	0.362	0.104	health,information,+care,...	116	145
Multiple	18	0.386	0.104	+analysis,+interface,data...	114	172
Multiple	19	0.390	0.104	+function,+macro,+progr...	109	182
Multiple	20	0.356	0.103	+entry,+frame,+develope...	108	145
Multiple	21	0.341	0.097	+entry,+catalog,+catalog ...	77	91
Multiple	22	0.315	0.098	+year,institute,+presentat...	121	147
Multiple	23	0.264	0.095	+programmer,+statemen...	98	163
Multiple	24	0.259	0.095	+program,+development,...	121	154
Multiple	25	0.255	0.092	+version,integration,+ent...	97	153

Note: If you ran the optional spell-checking step in "Using the Text Filter Node" on page 51, then the Topics shown here and those represented in subsequent steps might be different from what you see.

5. Select the Number of Documents by Topics chart to see a topic by the number of documents that it contains.



Note: You might need to resize the default graph to see the topic ID values.

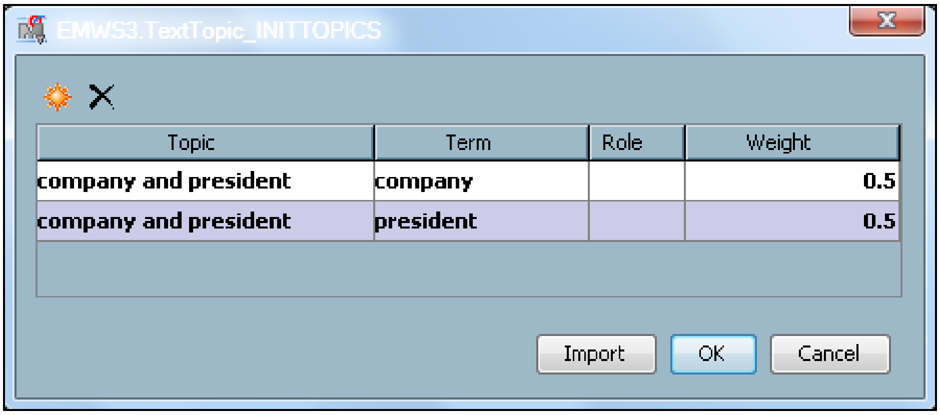
In addition to multi-term topics, you can use the **Text Topic** node to create single-term topics or to create your own topics.

6. Close the Results window, and select the **Text Topic** node.

7. Select the **Number of Single-term Topics** property, type *10*, and press **Enter** on your keyboard.

8. Click the ellipsis button for the **User Topics** property.

9. In the User Topics dialog box, click the **Add** button twice to create two rows. Enter the terms *company* and *president*, each with a weight of *0.5*, and specify the topic *company and president* for both.

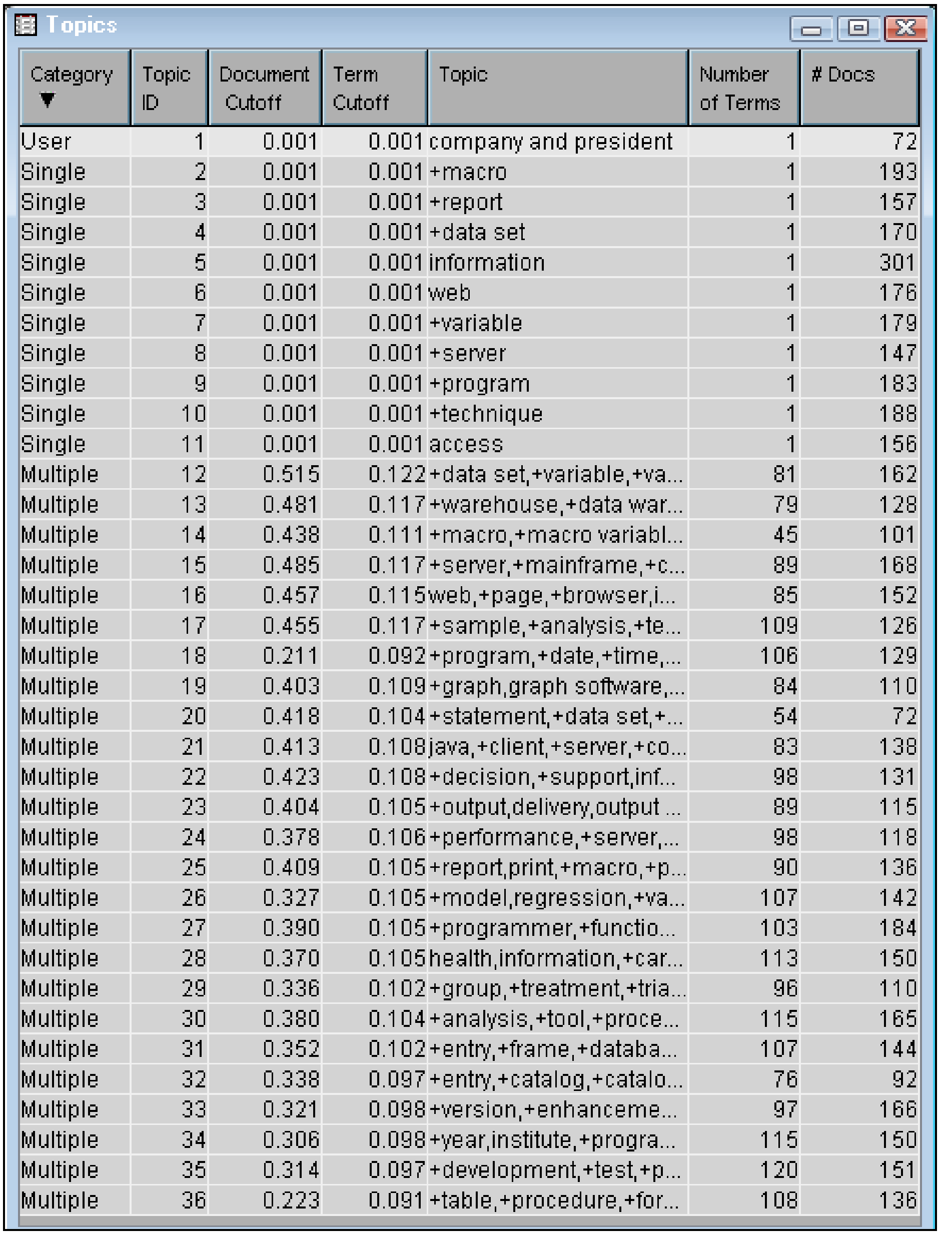


Topic	Term	Role	Weight
company and president	company		0.5
company and president	president		0.5

10. Click **OK**.

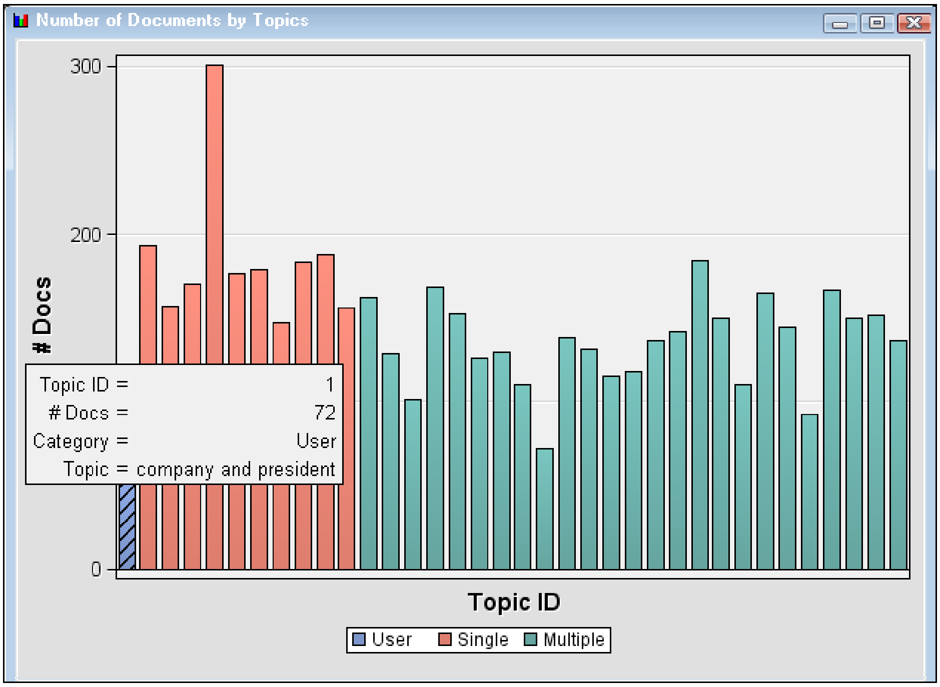
11. Right-click the **Text Topic** node and select **Run**. Select **Yes** in the Confirmation dialog box, and then **Results** in the Run Status dialog box when the node finishes running.

12. Select the Topics table. Notice that 10 new single-term topics have been created along with the topic that you specified in the User Topics dialog box.



Category ▼	Topic ID	Document Cutoff	Term Cutoff	Topic	Number of Terms	# Docs
User	1	0.001	0.001	company and president	1	72
Single	2	0.001	0.001	+macro	1	193
Single	3	0.001	0.001	+report	1	157
Single	4	0.001	0.001	+data set	1	170
Single	5	0.001	0.001	information	1	301
Single	6	0.001	0.001	web	1	176
Single	7	0.001	0.001	+variable	1	179
Single	8	0.001	0.001	+server	1	147
Single	9	0.001	0.001	+program	1	183
Single	10	0.001	0.001	+technique	1	188
Single	11	0.001	0.001	access	1	156
Multiple	12	0.515	0.122	+data set,+variable,+va...	81	162
Multiple	13	0.481	0.117	+warehouse,+data war...	79	128
Multiple	14	0.438	0.111	+macro,+macro variabl...	45	101
Multiple	15	0.485	0.117	+server,+mainframe,+c...	89	168
Multiple	16	0.457	0.115	web,+page,+browser,i...	85	152
Multiple	17	0.455	0.117	+sample,+analysis,+te...	109	126
Multiple	18	0.211	0.092	+program,+date,+time,...	106	129
Multiple	19	0.403	0.109	+graph,graph software,...	84	110
Multiple	20	0.418	0.104	+statement,+data set,+...	54	72
Multiple	21	0.413	0.108	java,+client,+server,+co...	83	138
Multiple	22	0.423	0.108	+decision,+support,inf...	98	131
Multiple	23	0.404	0.105	+output,delivery,output ...	89	115
Multiple	24	0.378	0.106	+performance,+server,...	98	118
Multiple	25	0.409	0.105	+report,print,+macro,+p...	90	136
Multiple	26	0.327	0.105	+model,regression,+va...	107	142
Multiple	27	0.390	0.105	+programmer,+functio...	103	184
Multiple	28	0.370	0.105	health,information,+car...	113	150
Multiple	29	0.336	0.102	+group,+treatment,+tria...	96	110
Multiple	30	0.380	0.104	+analysis,+tool,+proce...	115	165
Multiple	31	0.352	0.102	+entry,+frame,+databa...	107	144
Multiple	32	0.338	0.097	+entry,+catalog,+catalo...	76	92
Multiple	33	0.321	0.098	+version,+enhanceme...	97	166
Multiple	34	0.306	0.098	+year,institute,+progra...	115	150
Multiple	35	0.314	0.097	+development,+test,+p...	120	151
Multiple	36	0.223	0.091	+table,+procedure,+for...	108	136

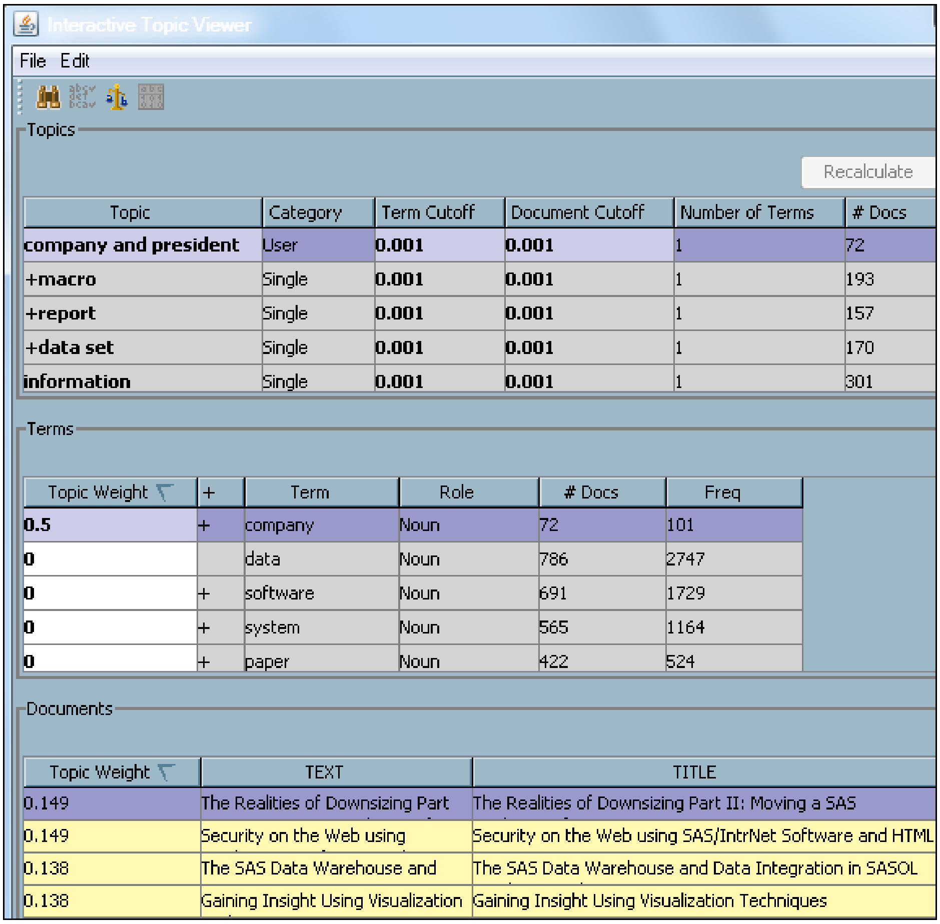
13. Select the Number of Documents by Topics window to see the multi-term, single-term, and user-created topics by the number of documents that they contain.



Topic ID = 1
# Docs = 72
Category = User
Topic = company and president

You can use the Interactive Topic Viewer to view and modify topic properties.

14. Close the Results window, and select the **Text Topic** node. Click the ellipsis button for the **Topic Viewer** property.



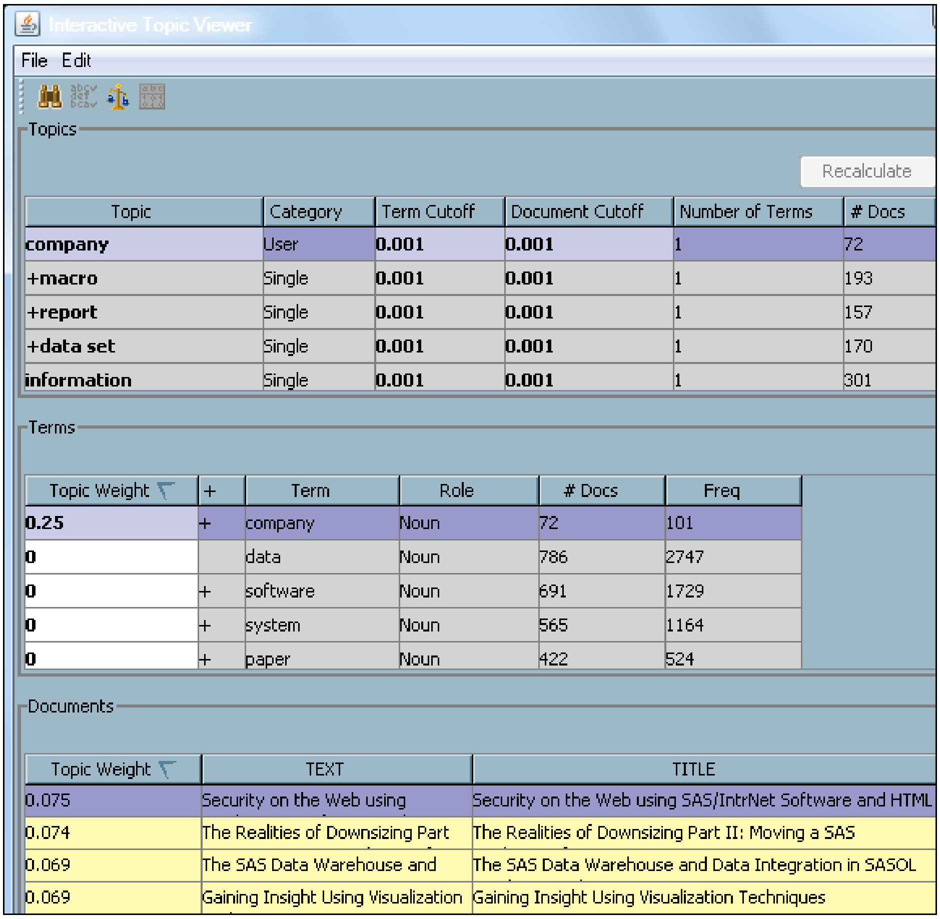
Topic	Category	Term Cutoff	Document Cutoff	Number of Terms	# Docs
company and president	User	**0.001**	**0.001**	1	72
+macro	Single	**0.001**	**0.001**	1	193
+report	Single	**0.001**	**0.001**	1	157
+data set	Single	**0.001**	**0.001**	1	170
information	Single	**0.001**	**0.001**	1	301

Topic Weight	+	Term	Role	# Docs	Freq
0.5	+	company	Noun	72	101
0		data	Noun	786	2747
0	+	software	Noun	691	1729
0	+	system	Noun	565	1164
0	+	paper	Noun	422	524

Topic Weight	TEXT	TITLE
0.149	The Realities of Downsizing Part	The Realities of Downsizing Part II: Moving a SAS
0.149	Security on the Web using	Security on the Web using SAS/IntrNet Software and HTML
0.138	The SAS Data Warehouse and	The SAS Data Warehouse and Data Integration in SASOL
0.138	Gaining Insight Using Visualization	Gaining Insight Using Visualization Techniques

In the Interactive Topic Viewer, you can change the topic name, term and document cutoff values, and the topic weight.

15. Select the topic value “company and president” in the Topics table and rename the topic to *company*. Select the topic weight for the term “company” in the Terms table, and change it to *0.25*. Click **Recalculate**.



Topic	Category	Term Cutoff	Document Cutoff	Number of Terms	# Docs
company	User	0.001	0.001	1	72
+macro	Single	0.001	0.001	1	193
+report	Single	0.001	0.001	1	157
+data set	Single	0.001	0.001	1	170
information	Single	0.001	0.001	1	301

Topic Weight	+	Term	Role	# Docs	Freq
0.25	+	company	Noun	72	101
0		data	Noun	786	2747
0	+	software	Noun	691	1729
0	+	system	Noun	565	1164
0	+	paper	Noun	422	524

Topic Weight	TEXT	TITLE
0.075	Security on the Web using	Security on the Web using SAS/IntrNet Software and HTML
0.074	The Realities of Downsizing Part	The Realities of Downsizing Part II: Moving a SAS
0.069	The SAS Data Warehouse and	The SAS Data Warehouse and Data Integration in SASOL
0.069	Gaining Insight Using Visualization	Gaining Insight Using Visualization Techniques

16. Close the Interactive Topic Viewer, and select **No** when prompted for whether you want to save your changes. For more information about the Interactive Topic Viewer, see “Interactive Topic Viewer” on page 95.

User-Defined Topic Lists

You can use a user-defined topic list in the **Text Topic** node. These lists enable you to define your own topics of interest. A sample user topic list is provided for use when analyzing the SAMPSIO.abstract sample data set.

User-defined topic data sets have a required format. You must include the variables **`_topic_`**, **`_term_`**, **`_role_`**, and **`_weight_`**. The variables **`_topic_`** and **`_weight_`** must have nonmissing values for all observations in the data set. Also, no observations can have blank values for both **`_term_`** and **`_role_`**.

- **`_topic_`** contains a unique identifier for each topic.

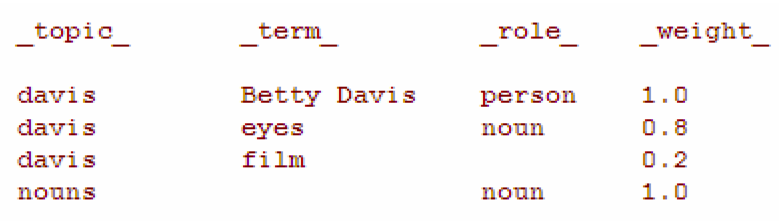
- **_term_** contains a term that is used in the topic identified by the value of **_topic_**. If this value is blank, then all terms that match the given role are assigned to the topic. For example, you could have a “People” topic that has one row with **_term_** blank, and **_role_** set to “PERSON”. The values in **_term_** are not case sensitive.

- **_role_** contains the role of **_term_**. A role is either a part of speech, an entity classification, or the value Noun Group. If this value is blank, then any time the **_term_** occurs with any role, it is considered to be in the topic. The values in **_role_** are not case sensitive. For more information about roles, see “Term Roles and Attributes” on page 36.

- **_weight_** contains the weight of term and role pair. Weights are relative. Give most important term and role pairs a weight of **1** and less important term and role pairs positive weights less than one in order to reflect their relative importance. Note that terms with a **_weight_** of **0** are omitted.

Several terms can be used in the same topic. To define topics with several terms, include multiple observations that all have the same value of **_topic_**. Each observation corresponds to a different term.

For example, the following user-defined topic list defines two topics (“davis” and “nouns”). The nouns topic would include all terms with a role of Noun:



topic	_term_	_role_	_weight_
davis	Betty Davis	person	1.0
davis	eyes	noun	0.8
davis	film		0.2
nouns		noun	1.0

Interactive Topic Viewer

Contents

Overview of the Interactive Topic Viewer

The Interactive Topic Viewer enables you to refine the topics that were generated (either automatically or from user-defined topics) when the **Text Topic** node was run. You can edit values that appear in bold in the tables of the viewer. You can save any changes that you make to the topics when you close the Interactive Topic Viewer. Any change to a topic name, cutoff, or any topic weights will cause that topic to be a User topic and to be stored in the data set in the User Topics property. To access the Interactive Topic Viewer, click the ellipsis button that corresponds to the **Topic Viewer** property in the Properties panel of the **Text Topic** node.

The Interactive Topic Viewer initially opens as a window that is split three ways: at the top is the Topics table, which contains a list of all user, single, and multi-term topics. One of these topics is always selected as the active topic. When the viewer is first opened, this is the first topic in the topic table. Next is the Terms table, which contains the weights for the selected topic for each term sorted by descending weight. Last is the Documents table, which contains topic weights for the selected topic also sorted by descending weight. For easier viewing, any one of these tables can be hidden by clicking on the arrow icon at the top right of any of the tables. Clicking that icon a second time opens that table, or clicking to hide any other table causes the new table to be hidden and the old one to reappear. Any of the columns in any of the tables can be resized or dragged to move, and any column can be sorted by clicking on it. Clicking the same column again causes it to be sorted in the opposite order.

Topics Table of the Interactive Topic Viewer

The Topics table displays summary information about the generated topics. You can edit the following properties of a topic:

- topic name
- term cutoff value
- document cutoff value

Note: The topic name, by default, are those five terms that have the highest topic weights in the Terms table. However, you can have a more succinct heading that you would like to use here.

The Terms table and Documents table are linked to the Topics table. You can select a different topic by either double-clicking an observation in the Topics table, or by right-clicking on a topic and choosing **Select Current Topic**. Then the Terms and Document tables will update to display values related to the selected topic. Furthermore, if you edit values in the Topics table, then you must click **Recalculate** to repopulate these two tables with the new information that results from the edit.

Terms Table of the Interactive Topic Viewer

The Terms table displays summary information about the terms that formulate the topic selected in the Topics table. Terms with weights that are higher in absolute value than the term cutoff contribute to the particular topic; those that are lower do not contribute to the topic. You can edit the topic weight for any term. If you edit the topic weight for a term, then the topic is automatically reclassified as a user-specified topic, if it was not classified as such already.

Documents Table of the Interactive Topic Viewer

The Documents table displays the relation of each document to the selected topic. If the topic weight for the document is greater than the document cutoff for the topic, the document is considered to contain the topic. This information includes, in addition to the topic weight for a document, all of the variables in the input data source. The second variable that is displayed is the variable that was used for text parsing. To wrap or unwrap the full contents of this variable in the column cells, right-click the value for an observation and select **Toggle Show Full Text** from the resulting menu.

The Interactive Topic Viewer enables you to examine each document on a term-by-term basis. In the Documents table, right-click the document that you want to investigate and select **Show Document Terms**, or click on the **Show Document Terms** icon in the top left. This refreshes the Terms table so that it contains only the terms in the document that you selected. Selecting a new document in the Documents table updates the Terms table to include only the terms from that document. To go back to showing all terms, right-click and choose **Show All Terms** or click on the **Show All Terms** icon in the top left.

Merging Topics

To merge multiple topics into one, enter the same topic name for all topics to be merged. They will initially still show up as distinct topics, but when you rerun the node, they will be combined into one, with all terms included from all topics. For all terms in common between the topics, the weight will be the average for each of the topics.

Chapter 8

The Text Rule Builder Node

Overview of the Text Rule Builder Node

The **Text Rule Builder** node generates an ordered set of rules that together are useful in describing and predicting a target variable. Each rule in the set is associated with a specific target category, consisting of a conjunction that indicates the presence or absence of one or a small subset of terms (for example, “term1” AND “term2” AND (NOT “term3”)). A particular document matches this rule if and only if it contains at least one occurrence of term1 and of term2 but no occurrences of term3.

This set of derived rules creates a model that is both descriptive and predictive. When categorizing a new document, it will proceed through the ordered set and choose the target that is associated with the first rule that matches that document. The rules are provided in the syntax that can be used within SAS Content Categorization Studio, and can be deployed there.

The **Text Rule Builder** node is a standard SAS Enterprise Miner modeling tool, complete with the standard reporting features. You can view which predicted target values are most likely to be wrong based on the generated model. Optionally, you can change the target that is assigned to some of these observations and rerun the results. Thus, it facilitates “active learning” in which a user can dynamically interact with an algorithm to iteratively build a predictive model.

See the following for more information about the **Text Rule Builder** node:

- “Text Rule Builder Node Input Data” on page 100
- “Text Rule Builder Node Properties” on page 100
- “Text Rule Builder Node Results” on page 104
- “Text Rule Builder Node Output Data” on page 103
- “Using the Text Rule Builder Node” on page 106

Text Rule Builder Node Input Data

The training data that is provided to the **Text Rule Builder** node must be preceded by at least one **Text Parsing** and one **Text Filter** node. The node must have at least one target variable with a measurement level of binary, ordinal, or nominal. Finally, the **Drop** property for the target variable must be set to **No**. If any of these conditions are not met, the **Text Rule Builder** node will generate an error.

If you plan on using validation or test data, those data sets must contain the same target variables that are in the training data.

Text Rule Builder Node Properties

Contents

Text Rule Builder Node General Properties

These are the general properties that are available on the **Text Rule Builder** node:

- **Node ID** — displays the ID that is assigned to the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Rule Builder** node that is added to a diagram has the Node ID `TextRule`, and the second **Text Rule Builder** node that is added has the Node ID `TextRule2`.

- **Imported Data** — accesses a list of the data sets that are imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do any of the following:

  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set

- **Exported Data** — accesses a list of the data sets that are exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do any of the following:
  - browse the data set
  - explore (sample and plot) the data in a data set
  - view the table and variable properties of a data set
- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Rule Builder Node Train Properties

These are the training properties that are available on the **Text Rule Builder** node:

- **Variables** — choose which categorical target to use in this node (note that only one target can be analyzed). Click the ellipsis button to open the Variables window.
- **Generalization Error** — determines the predicted probability for rules that use an untrained data set. This is to prevent overtraining. Higher values do a better job of preventing overtraining at a cost of not finding potentially useful rules. Valid values are **Very Low**, **Low**, **Medium** (default), **High**, and **Very High**.
- **Purity of Rules** — determines how selective each rule is by controlling the maximum p-value necessary to add a term to a rule. Selecting **Very High** results in the fewest, purest rules. Selecting **Very Low** results in the most rules that handle the most terms. Valid values are **Very Low** (p<.17), **Low** (p<.05), **Medium** (default, p<. 005), **High** (p<.0005), and **Very High** (p<.00005).
- **Exhaustiveness** — determines the exhaustiveness of the rule search process, or how many potential rules are considered at each step. As you increase the exhaustiveness, you increase the amount of time that the **Text Rule Builder** node requires and increase the probability of overtraining the model. Valid values are **Very Low**, **Low**, **Medium** (default), **High**, and **Very High**.

Text Rule Builder Node Score Properties

These are the score properties that are available on the **Text Rule Builder** node:

- **Content Categorization Code** — Select the ellipsis button to the right of the **Content Categorization Code** property to view the Content Categorization Code window. The code that is provided in this window can be copied and pasted into SAS Content Categorization Studio. The **Text Rule Builder** node must be run before you can open the Content Categorization Code window.

- **Change Target Values** — Select the ellipsis button to the right of the **Change Target Values** property to view the Change Target Values window. The Change Target Values window enables you to view and reassign target values. As a result, you can rerun the **Text Rule Builder** node and iteratively refine your model.

  The observations in the Change Target Values window contain all observations in the training, validation, or test data set that meet any of the following conditions:

  - contains a rule that predicted a target value other than the target assigned to this document in the imported data. This includes observations where the target contains a missing value.

  - an observation where you have previously changed the imported target value to a different target value.

The observations in the Change Target Values window are ordered by the model’s determined “posterior probability” in descending order from 1 to 0. Therefore, the values that the model is most certain are incorrect are at the very beginning.

The data set that is shown in the Change Target Values window is not created until you run the node, and the node will generate an error if you try to view the Change Target Values window before running the node. Any changes to the assigned target value are retained and used when the node is rerun, as long as the target variable has not been changed. When you rerun the node, your changes are applied to the data before the rule creation algorithm is run.

If you copy a **Text Rule Builder** node, then the **Change Target Values** data set is copied to the new node.

Text Rule Builder Node Status Properties

These are the status properties that are displayed on the **Text Rule Builder** node:

- **Create Time** — time that the node was created.

- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.

- **Last Error** — error message, if any, from the last run.

- **Last Status** — last reported status of the node.

- **Last Run Time** — time at which the node was last run.

- **Run Duration** — length of time required to complete the last node run.

- **Grid Host** — grid host, if any, that was used for computation.

- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner extension node. The value of this property is always `No` for the **Text Rule Builder** node.

Text Rule Builder Node Output Data

The **Text Rule Builder** node exports the following:

- **Train Role** — contains the imported training data set to be imported to an assessment tool with the following additional columns:

Role	Form	Type	Source	Example Name	Purpose
CLASSIFICATION	F_<target>	$32	Decmeta, Type=FROM	F_c_target	Formatted target value normalized to a length of 32.
CLASSIFICATION	I_<target>	$32	Decmeta, Type=INTO	I_c_target	Formatted predicted target value normalized to a length of 32.
PREDICTED	P_<target> <target-value>	8	Decmeta, Type=PREDICTED,level=formatted normalized target value	P_c_targethockey	There is one of these for every target value. This contains the posterior probability generated by the model that the observation belongs to for that target. The sum of these should equal one for every observation.
ASSESSMENT	W_<target>	8		W_c_target	Why the observation was assigned the INTO value that it was. This contains the number of the rule that triggered the classification, or missing if no rule matched and it was assigned by default to the most common remaining value.

- **Validate, Test Role** — contains any imported validation, or test data set to be imported to an assessment tool with the same additional columns as listed above.
- **EMINFO** — contains the following:
  - key=LastTmNode, Value=&EM_NODEID
  - key=LastTMNodeType, Value=TextBoolCat
  - key=LastTextBoolCat, value=&EM_NODEID
  - key=PRESCORECODE, value=&EM_NODEID

See "Output Data for SAS Text Miner Nodes" on page 127 for more information.

Text Rule Builder Node Results

Contents

Results Window for the Text Rule Builder Node

After the **Text Rule Builder** node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status window that opens immediately after a successful run of the **Text Rule Builder** node.
- Click the Results icon on the main toolbar.
- Right-click the **Text Rule Builder** node, and select **Results**.

The Results window for the **Text Rule Builder** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are the same as other nodes. However, the **View** menu for the **Text Rule Builder** node Results window does not include the selection **Model**. Instead, it includes **Rules**, which accesses a submenu that lists the generated rules in tabular form.

Note: You can access the SAS log that was generated by the node processing from the **SAS Results** submenu of the **View** menu. This log can be a useful debugging tool.

Text Rule Builder Node Graphical and Tabular Results

The following are the graphical results in the **Text Rule Builder** node Results window:

- **Score Rankings Overlay** — The Score Rankings Overlay chart displays assessment statistics. To change the graphed statistic, select one of the following items from the drop-down menu:

  - **Cumulative Lift**
  - **Lift**
  - **Gain**
  - **% Response**
  - **Cumulative % Response**
  - **% Captured Response**
  - **Cumulative % Captured Response**

  For more graphical options, right-click the Score Rankings Overlay chart, and select **Data Options**.

- **Fit Statistics** — The Fit Statistics table displays the following statistics for the target variable:
  - Average Squared Error
  - Divisor for ASE
  - Maximum Absolute Error
  - Sum of Frequencies
  - Root Average Squared Error
  - Sum of Squared Errors
  - Frequency of Classified Cases
  - Misclassification Rate
  - Number of Wrong Classifications

- **Rules Obtained** — The Rules Obtained table displays rules for predicting the target variable.

  The Rule column contains the rules that are extracted from the text. These rules are presented as the conjunction of terms and their negations. For example, the Rule "dividend&~acquire&~sell" says that for a document to satisfy this rule, it must contain the term “dividend” and should not contain the terms “acquire” and “sell”. Note that the '~' character is the negation operation on a term. If a document satisfies a rule, the document is covered by that rule. Otherwise, it is uncovered.

  The “Target Value” column indicates the target value that corresponds to the rule. It indicates that if a document satisfies that rule, this document should be assigned the particular target value. The order of the rules in the table is important. The rule in the first row of the table is discovered by considering all the documents and is the first rule that is added into the rule set. The rule in the second row of the table is learned by analyzing all documents that were not covered by the first rule, and so on. When the rules are applied to new data for scoring, it is assumed that they will be applied in this same order.

  Let r_i be the i^{th} rule that is added into the rule set. Then the **Remaining Positive/ Total** column of r_i indicates how many documents in the training data do not satisfy any of its previous rules r_1 ... , r_{i-1}, and how many of them are true positive. The **True Positive/Total** column of r_i indicates how many of the documents in the training data that is uncovered by its previous rules satisfy r_i, and among them how many are true positive.

The sample precision column and sample recall column of r_i contain the precision and recall of the rule set r_1 ... , r_i, which is computed using the training data. The estimated precision gives an indication of how well this rule is expected to work in holdout data, based on the generalization error property setting.

Two additional charts are available by navigating to **View** ⇨ **Assessment**:

- Select **Classification Chart** to display a stacked bar chart of the classification results for a categorical variable. The horizontal axis displays the target levels that observations actually belong to. The color of the stacked bars identifies the target levels that observations are classified into. The height of the stacked bars represents the percentage of total observations.

- Select **Score Distribution** to display a chart that plots the proportion of events (by default), nonevents, and other values on the vertical axis. The values on the horizontal axis represent the model score of a bin. The model score depends on the prediction of the target. For categorical targets, observations are grouped into bins, based on the posterior probabilities of the event level and the number of buckets. The Score Distribution chart of a useful model shows a higher percentage of events for higher model score and a higher percentage of nonevents for lower model scores. The chart choices are as follows:

  - **Percentage of Events** — for categorical targets.
  - **Number of Events** — for categorical targets.
  - **Cumulative Percentage of Events** — for categorical targets.

For more information about graphical and tabular results in SAS Enterprise Miner, see the Predictive Modeling Help topic in the SAS Enterprise Miner help.

Text Rule Builder Node SAS Output Results

The SAS output from the **Text Rule Builder** node includes summary information about the input variables, targets, predicted and decision variables, training results, and validation results (if a **Data Partition** node was used). The Classification Table contains precision (as Target Percentage) and recall (as Outcome Percentage), for both the training and validation data.

Using the Text Rule Builder Node

This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

The **Text Rule Builder** node creates Boolean rules from small subsets of terms to predict a categorical target variable. The node must be preceded by **Text Parsing** and **Text Filter** nodes.

This example uses the SAMPSIO.NEWS data set to show you how to predict a categorical target variable with the **Text Rule Builder** node. The results will also show that the model is highly interpretable and useful for explanatory and summary purposes as well.

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

The SAMPSIO.NEWS data set contains 600 observations and the following variables:

- **TEXT** is a nominal variable that contains the text of the news article.
- **graphics** is a binary variable that indicates whether the document belongs to the computer graphics category (1-yes, 0-no).
- **hockey** is a binary variable that indicates whether the document belongs to the hockey category (1-yes, 0-no).
- **medical** is a binary variable that indicates whether the document is related to medical issues (1-yes, 0-no).
- **newsgroup** is a nominal variable that contains the group that a news article fits into.

To use the **Text Rule Builder** node to predict the categorical target variable, **newsgroup**, in the SAMPSIO.NEWS data set:

1. Use the Data Source Wizard to define a data source for the data set SAMPSIO.NEWS.

   a. Set the measurement levels of the variables **graphics**, **hockey**, and **medical** to **`Binary`**.

   b. Set the model role of the variable **newsgroup** to **`Target`** and leave the roles of **graphics**, **hockey**, and **medical** as **`Input`**.

   c. Set the variable **text** to have a role of **`Text`**.

   d. Select **`No`** in the Data Source Wizard — Decision Configuration dialog box.

   e. Use the default target profile for the target **newsgroup**.

2. After you create the **NEWS** data source, drag it to the diagram workspace.

3. Select the **Text Mining** tab on the toolbar, and drag a **Text Parsing** node into the diagram workspace.

4. Connect the **NEWS** data source to the **Text Parsing** node.

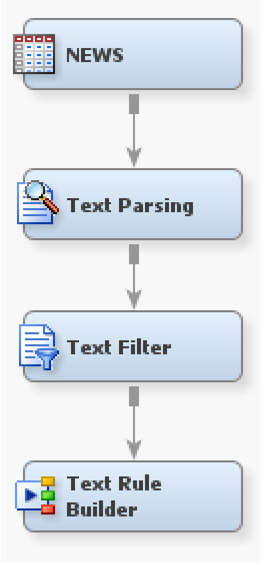
5. Select the **Text Mining** tab on the toolbar, and drag a **Text Filter** node into the diagram workspace.

6. Connect the **Text Parsing** node to the **Text Filter** node.

7. Select the **Text Mining** tab on the toolbar, and drag a **Text Rule Builder** node into the diagram workspace.

8. Connect the **Text Filter** node to the **Text Rule Builder** node.

   Your process flow diagram should resemble the following:



9. Select the **Text Rule Builder** node in the process flow diagram.

10. Click the value for the **Generalization Error** property, and select `Very Low`.

11. Click the value for the **Purity of Rules** property, and select `Very Low`.

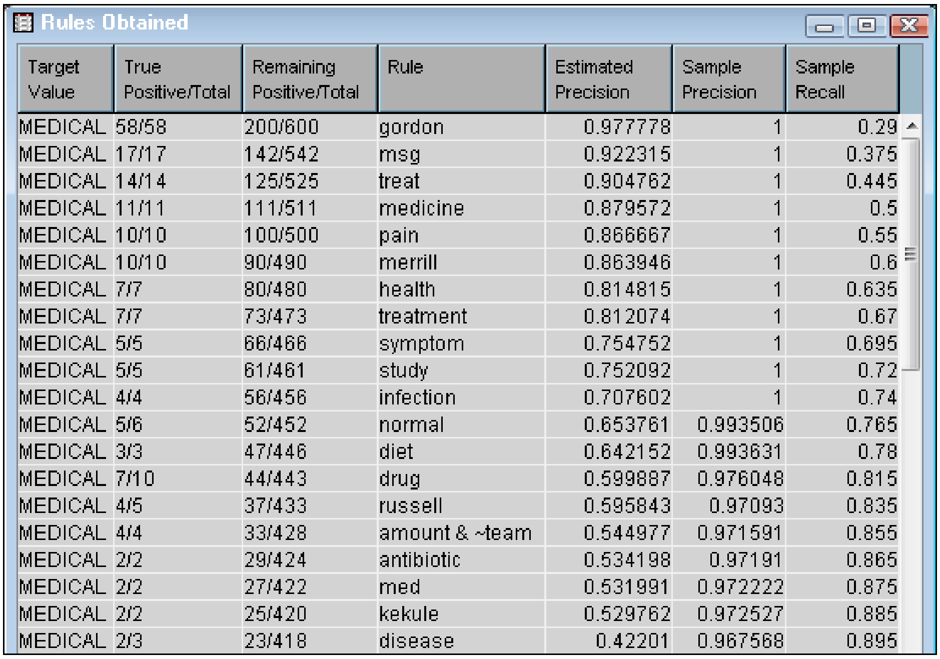
12. Click the value for the **Exhaustiveness** property, and select `Very Low`.

13. In the diagram workspace, right-click the **Text Rule Builder** node and select **Run**. Click **Yes** in the Confirmation dialog box that appears.

14. Click **Results** in the Run Status dialog box when the node finishes running.

15. Select the Rules Obtained table to see information about the rules that were obtained.

    The words in the Rule column have the corresponding estimated precision at implying the target, **newsgroup**.



Target Value	True Positive/Total	Remaining Positive/Total	Rule	Estimated Precision	Sample Precision	Sample Recall
MEDICAL	58/58	200/600	gordon	0.977778	1	0.29
MEDICAL	17/17	142/542	msg	0.922315	1	0.375
MEDICAL	14/14	125/525	treat	0.904762	1	0.445
MEDICAL	11/11	111/511	medicine	0.879572	1	0.5
MEDICAL	10/10	100/500	pain	0.866667	1	0.55
MEDICAL	10/10	90/490	merrill	0.863946	1	0.6
MEDICAL	7/7	80/480	health	0.814815	1	0.635
MEDICAL	7/7	73/473	treatment	0.812074	1	0.67
MEDICAL	5/5	66/466	symptom	0.754752	1	0.695
MEDICAL	5/5	61/461	study	0.752092	1	0.72
MEDICAL	4/4	56/456	infection	0.707602	1	0.74
MEDICAL	5/6	52/452	normal	0.653761	0.993506	0.765
MEDICAL	3/3	47/446	diet	0.642152	0.993631	0.78
MEDICAL	7/10	44/443	drug	0.599887	0.976048	0.815
MEDICAL	4/5	37/433	russell	0.595843	0.97093	0.835
MEDICAL	4/4	33/428	amount & ~team	0.544977	0.971591	0.855
MEDICAL	2/2	29/424	antibiotic	0.534198	0.97191	0.865
MEDICAL	2/2	27/422	med	0.531991	0.972222	0.875
MEDICAL	2/2	25/420	kekule	0.529762	0.972527	0.885
MEDICAL	2/3	23/418	disease	0.42201	0.967568	0.895

In the second column above, the True Positive (the first number) is the number of documents that were correctly assigned to the rule. The Total (the second number) is the total positive.

In the third column above, the Remaining Positive (the first number) is the total number of remaining documents in the category. The Total (the second number) is the total number of documents remaining.

In the above example, in the first row, 200 documents have been assigned to the MEDICAL newsgroup, and 600 total documents exist in the data set. 58 of the documents were assigned to the rule "gordon" (58 were correctly assigned). This means that if a document contains the word "gordon," and you assign all those documents to the MEDICAL newsgroup, 58 out of 58 will be assigned correctly. In the next row, there are 200 – 58 = 142 MEDICAL newsgroup documents left that can be evaluated for rule assignment, out of a total of 600 – 58 = 542 documents. In this second row, 17 documents are correctly assigned to the rule "msg." This means that if a document contains the term "msg," and you assign all those documents to the MEDICAL newsgroup, 17 out of 17 will be assigned correctly.

Most of the rules are single term rules because the NEWS data set is limited in size. However, there is one multiple term rule above. In the 16th row, the rule "amount & ~team" means that if a document contains the word "amount" and does not contain the word "team," then 4 of the remaining documents will be correctly assigned to the MEDICAL newsgroup.

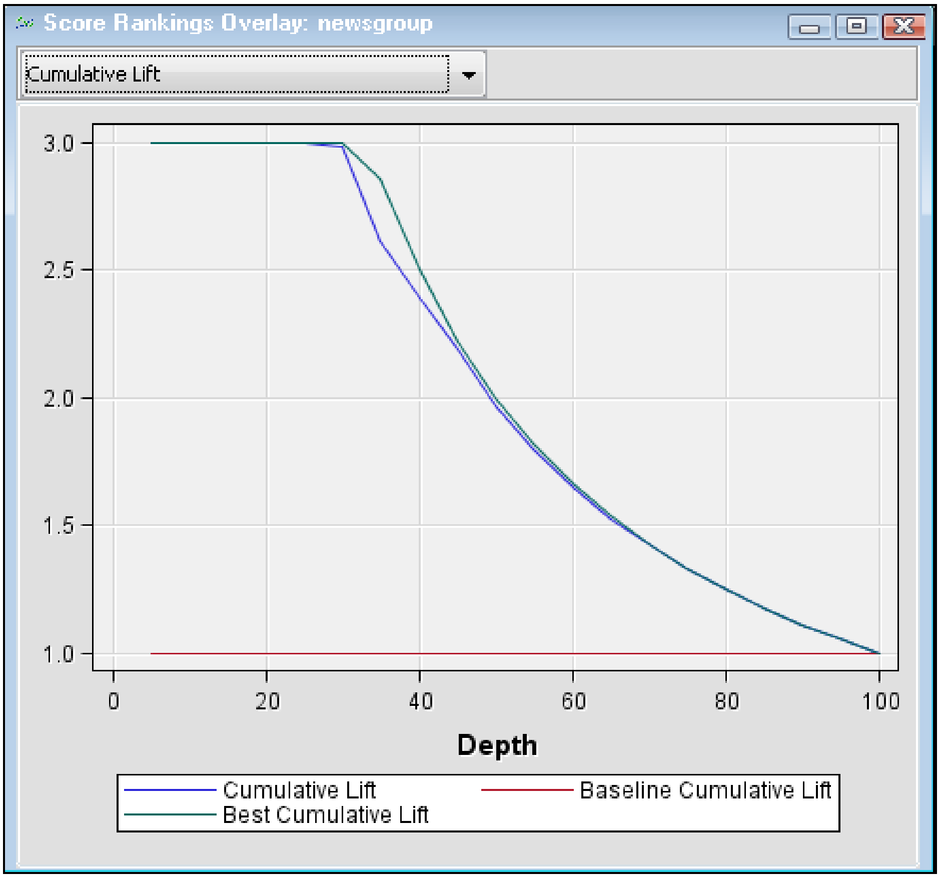
Note: ~ means logical not.

16. Select the Score Rankings Overlay graph to view the following types of information about the target variable:

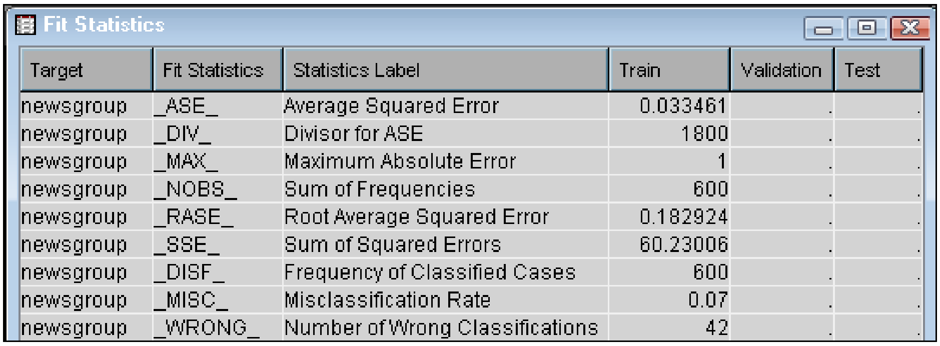
- Cumulative Lift
- Lift
- Gain
- % Response
- Cumulative % Response

- % Captured Response
- Cumulative % Captured Response

Note: To change the statistic, select one of the above options from the drop-down menu.



17. Select the Fit Statistics window for statistical information about the target variable, **newsgroup**.



Target	Fit Statistics	Statistics Label	Train	Validation	Test
newsgroup	_ASE_	Average Squared Error	0.033461	.	.
newsgroup	_DIV_	Divisor for ASE	1800	.	.
newsgroup	_MAX_	Maximum Absolute Error	1	.	.
newsgroup	_NOBS_	Sum of Frequencies	600	.	.
newsgroup	_RASE_	Root Average Squared Error	0.182924	.	.
newsgroup	_SSE_	Sum of Squared Errors	60.23006	.	.
newsgroup	_DISF_	Frequency of Classified Cases	600	.	.
newsgroup	_MISC_	Misclassification Rate	0.07	.	.
newsgroup	_WRONG_	Number of Wrong Classifications	42	.	.

18. Close the Results window.

19. Click the value for the **Generalization Error** property, and select `Medium`.

20. Click the value for the **Purity of Rules** property, and select `Medium`.

21. Click the value for the **Exhaustiveness** property, and select `Medium`.

22. Select the **NEWS** data source.

23. Click the ellipsis button for the **Variables** property.

24. Change the role of the HOCKEY variable to **`Target`**, and change the role of the NEWSGROUP variable to **`Input`**.

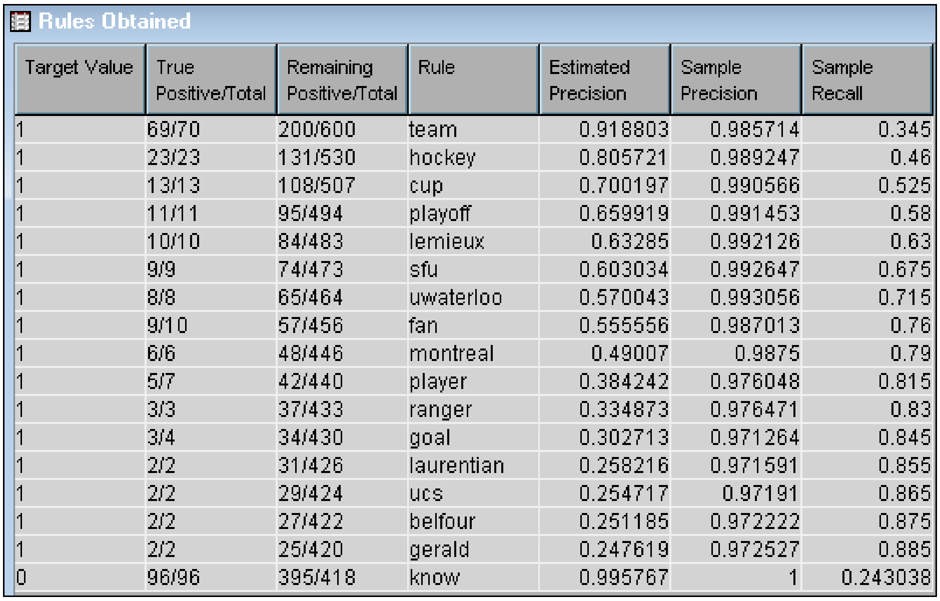
25. Click **OK**.

26. In the diagram workspace, right-click the **Text Rule Builder** node and select **Run**. Click **Yes** in the Confirmation dialog box that appears.

27. Click **Results** in the Run Status dialog box when the node finishes running.

28. Select the Rules Obtained table to see information about the rules that predicted the target — the HOCKEY newsgroup.

    The words in the Rule column have the corresponding estimated precision at implying the hockey target.



Rules Obtained

Target Value	True Positive/Total	Remaining Positive/Total	Rule	Estimated Precision	Sample Precision	Sample Recall
1	69/70	200/600	team	0.918803	0.985714	0.345
1	23/23	131/530	hockey	0.805721	0.989247	0.46
1	13/13	108/507	cup	0.700197	0.990566	0.525
1	11/11	95/494	playoff	0.659919	0.991453	0.58
1	10/10	84/483	lemieux	0.63285	0.992126	0.63
1	9/9	74/473	sfu	0.603034	0.992647	0.675
1	8/8	65/464	uwaterloo	0.570043	0.993056	0.715
1	9/10	57/456	fan	0.555556	0.987013	0.76
1	6/6	48/446	montreal	0.49007	0.9875	0.79
1	5/7	42/440	player	0.384242	0.976048	0.815
1	3/3	37/433	ranger	0.334873	0.976471	0.83
1	3/4	34/430	goal	0.302713	0.971264	0.845
1	2/2	31/426	laurentian	0.258216	0.971591	0.855
1	2/2	29/424	ucs	0.254717	0.97191	0.865
1	2/2	27/422	belfour	0.251185	0.972222	0.875
1	2/2	25/420	gerald	0.247619	0.972527	0.885
0	96/96	395/418	know	0.995767	1	0.243038

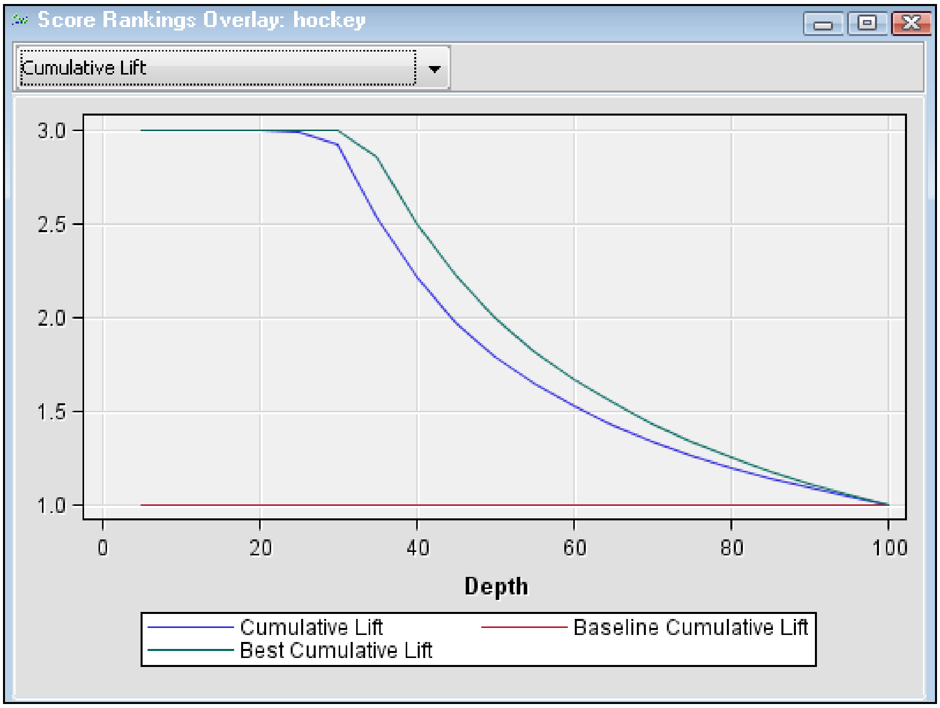
In the above example, in the first row, 200 documents have been assigned to the HOCKEY newsgroup, and 600 total documents exist in the data set. The target value is `1`, instead of "HOCKEY," because we set the **hockey** variable to be the target instead of the **newsgroup** variable. 70 of the documents were assigned to the rule "team" (69 were correctly assigned). This means that if a document contains the word "team," and you assign all those documents to the HOCKEY newsgroup, 69 out of 70 will be assigned correctly. In the next row, there are 200 – 69 = 131 HOCKEY documents left that can be evaluated for rule assignment, out of a total of 600 – 70 = 530 documents. In this second row, 23 documents are correctly assigned to the rule "hockey." This means that if a document contains the word "hockey," and you assign all those documents to the HOCKEY newsgroup, 23 out of 23 will be assigned correctly.

29. Select the Score Rankings Overlay graph to view the following types of information about the target variable:

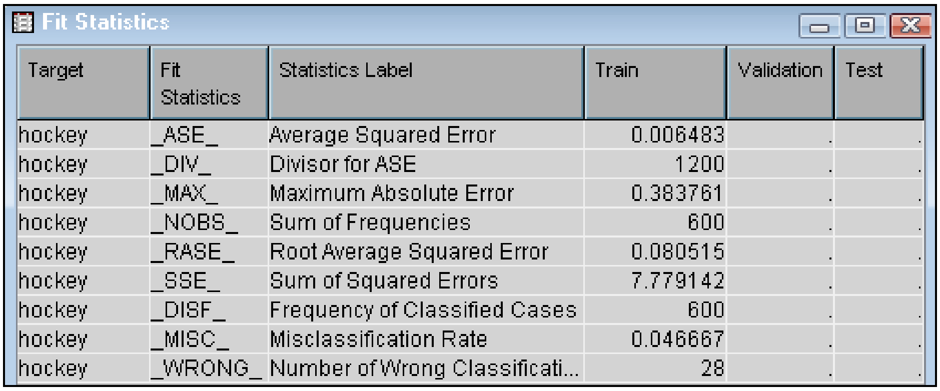
    - Cumulative Lift

- Lift
- Gain
- % Response
- Cumulative % Response
- % Captured Response
- Cumulative % Captured Response

Note: To change the statistic, select one of the above options from the drop-down menu.



30. Select the Fit Statistics table for statistical information about the hockey target variable.



Fit Statistics

Target	Fit Statistics	Statistics Label	Train	Validation	Test
hockey	_ASE_	Average Squared Error	0.006483	.	.
hockey	_DIV_	Divisor for ASE	1200	.	.
hockey	_MAX_	Maximum Absolute Error	0.383761	.	.
hockey	_NOBS_	Sum of Frequencies	600	.	.
hockey	_RASE_	Root Average Squared Error	0.080515	.	.
hockey	_SSE_	Sum of Squared Errors	7.779142	.	.
hockey	_DISF_	Frequency of Classified Cases	600	.	.
hockey	_MISC_	Misclassification Rate	0.046667	.	.
hockey	_WRONG_	Number of Wrong Classificati...	28	.	.

31. Close the Results window.

32. Click the ellipsis button for the **Content Categorization Code** property.

The Content Categorization Code window appears. The code provided in this window is the code that is output for SAS Content Categorization and is ready for compilation.

33. Click **Cancel**.

34. Click the ellipsis button for the **Change Target Values** property.

    The Change Target Values window appears.

    You can use the Change Target Values window to improve the model.

35. Select one or more cells in the **Assigned Target** column, and select a new target value.

36. Click **OK**.

37. Rerun the **Text Rule Builder** node, and then check whether your model has been improved.

Chapter 9
The Text Miner Node

Overview of the Text Miner Node

Note: The **Text Miner** node is not available from the **Text Mining** tab in SAS Text Miner 12.1. The **Text Miner** node has now been replaced by the functionality in other SAS Text Miner nodes. You can import diagrams from a previous release of SAS Text Miner that had a **Text Miner** node in the process flow diagram. However, new **Text Miner** nodes can no longer be created, and property values cannot be changed in imported **Text Miner** nodes. For more information, see "Converting SAS Text Miner Diagrams from a Previous Version" on page 11.

The **Text Miner** node enables you to discover and use the information that exists in a collection of documents. The node can process volumes of textual data such as e-mail messages, news articles, Web pages, research papers, and surveys, even if they are stored in data formats.

Data is processed in three phases: text parsing, transformation, and document clustering. Text parsing processes textual data into a term-by-document frequency matrix. Transformations such as singular value decomposition (SVD) alter this matrix into a data set that is suitable for data mining purposes; a document collection of several thousand documents and hundreds of thousands of terms can be represented in a compact and efficient form. Finally, the **Text Miner** node enables you to explore the text collection by clustering the documents, reporting on these clusters, and presenting visualizations of this data.

The **Text Miner** node can require a lot of computing time and resources to process a very large collection of documents. If your system has limited resources, try the following:

- Analyze only a sample of the document collection.
- Set some of theses properties to `No`: **Find Entities**, **Noun Groups**, **Terms in Single Document**.
- Reduce the number of requested SVD dimensions or roll-up terms.

  Note: If the **Text Miner** node encounters memory problems generating the SVD, then try rolling up a certain number of terms; the remaining terms are automatically dropped.
- To limit parsing to high-information words, disable all parts of speech except nouns, proper nouns, noun groups, and verbs.

The **Text Miner** node behaves like most nodes in SAS Enterprise Miner except that the **Text Miner** node does not generate portable score code. For more information about the **Text Miner** node, see the following:

- "Text Miner Node Input Data" on page 116
- "Text Miner Node Properties" on page 117
- "Text Miner Node Results" on page 121
- "Text Miner Node Output Data" on page 123

Text Miner Node Input Data

Note: The **Text Miner** node is not available from the **Text Mining** tab in SAS Text Miner 12.1. The **Text Miner** node has now been replaced by the functionality in other SAS Text Miner nodes. You can import diagrams from a previous release of SAS Text Miner that had a **Text Miner** node in the process flow diagram. However, new **Text Miner** nodes can no longer be created, and property values cannot be changed in imported **Text Miner** nodes. For more information, see "SAS Text Miner and SAS Session Encoding" on page 11.

The **Text Miner** node must be preceded by one or more Input Data Source nodes, where each data source contains a document collection to parse. At least one data source must have the role Train. Others can have roles of `Train`, `Valid`, `Test`, or `Score`.

Each observation in a data source represents an individual document in the document collection. The data source can have one of two structures. It can either contain the entire text of each document or it can contain paths to plain text or HMTL files that contain that text.

- If the data source contains the entire text of each document, then this text must be stored in a character variable that is assigned the role `Text`. Note that a SAS variable can hold only 32KB of text. If any document in the collection is larger than that limit, then you should not use this data source structure.

- There are two sample data sets that you can use to create data sources with this structure. For more information, see "SAS Text Miner Sample Data Sets" on page 9.

- If the data source contains paths to files that contain the document text, then these paths must be stored in a character variable that is assigned the role **`Text Location`**. The paths must be relative to the SAS Text Miner server. This structure can be used either for collections that contain smaller documents or documents that exceed the 32KB SAS variable limit.

  TIP

  - To help identify which link represents which document, you can include an additional character variable that contains truncated document text. Give this variable the role **`Text`**.

  - If there is a variable in the data set that contains the location of the unfiltered documents and if you assign this variable the role **`Web Address`**, you will be able access the original source of each document in the Interactive Results viewer.

In many cases, you will need to preprocess textual data before you can import it into a data source. A SAS macro named %TMFILTER is provided with SAS Text Miner. It can be used in file preprocessing to extract text from various document formats or to retrieve text from Web sites by crawling the Web. The macro creates a SAS data set that you can use to create a data source to use as input for the **Text Parsing** node. Depending on which structure (of the two described above) that you use, you will have to adjust the roles of the variables accordingly in the Data Source wizard.

A successful run of the **Text Miner** node requires at least one variable with the role **`Text`** or **`Text Location`** variable. If you have more than one variable with either role (that has a use status of **`Yes`**), then the longest of these variables is used.

Text Miner Node Properties

Contents

Note: The **Text Miner** node is not available from the **Text Mining** tab in SAS Text Miner 12.1. The **Text Miner** node has now been replaced by the functionality in other SAS Text Miner nodes. You can import diagrams from a previous release of SAS Text Miner that had a **Text Miner** node in the process flow diagram. However, new **Text Miner** nodes can no longer be created, and property values cannot be changed in imported **Text Miner** nodes. For more information, see "Converting SAS Text Miner Diagrams from a Previous Version" on page 11.

Text Miner Node General Properties

These are the general properties that are available on the **Text Miner** node:

- **Node ID** — displays the ID that SAS Enterprise Miner automatically assigns the node. Node IDs are especially useful for distinguishing between two or more nodes of the same type in a process flow diagram. For example, the first **Text Miner** node added to a diagram will have the node ID **`TEXT`**, and the second Text Miner node added will have the node ID **`TEXT2`**.

- **Imported Data** — accesses a list of the data sets imported by the node and the ports that provide them. Click the ellipsis button to open the Imported Data window, which displays this list. If data exists for an imported data set, then you can select a row in the list and do any of the following:

  - browse the data set

  - explore (sample and plot) the data in a data set

  - view the table and variable properties of a data set

- **Exported Data** — accesses a list of the data sets exported by the node and the ports to which they are provided. Click the ellipsis button to open the Exported Data window, which displays this list. If data exists for an exported data set, then you can select a row in the list and do any of the following:

  - browse the data set

  - explore (sample and plot) the data in a data set

  - view the table and variable properties of a data set

- **Notes** — accesses a window that you can use to store notes of interest, such as data or configuration information. Click the ellipsis button to open the Notes window.

Text Miner Node Train Properties

General Train Properties

These are the training properties that are available on the **Text Miner** node:

- **Variables** — accesses a list of variables and associated properties in the data source. Only variables with the roles Text, Text Location, or Web Address are displayed. Click the ellipsis button to open the Variables window.

- **Interactive** — This property is not available in SAS Text Miner 12.1.

- **Force Run** — specifies whether to rerun the node, even if it has already been successfully run. This function is useful (for example, if the underlying input data has changed).

Parse Properties

- **Parse Variable** — (value is populated after the node has run) displays the name of the variable in the input data source that was used for parsing. Depending on the structure of the data source, this variable contains either the entire text of each document in the document collection or it contains paths to plain text or HMTL files that contain that text.

- **Language** — specifies the language to use when parsing text. Only supported languages that are licensed to you are available for selection. For a list of supported languages, see "About SAS Text Miner" on page 4.

- **Stop List** — accesses a window in which you can select a SAS data set that contains terms to exclude from parsing. If you include a stop list, then the terms therein are not included in the node results. Default data sets are provided for several languages. You can edit these data sets or create your own. Click the ellipsis button to open the Select a SAS Table window.

- **Start List** — accesses a window in which you can select a SAS data set that contains the terms to parse. If you include a start list, then terms other than those therein are not included in the node results. Click the ellipsis button to open the Select a SAS Table window.

- **Stem Terms** — specifies whether to stem terms.

- **Terms in Single Document** — specifies whether to parse terms that appear in only one document.

- **Punctuation** — specifies whether to parse punctuation marks as terms.

- **Numbers** — specifies whether to parse numbers as terms.

- **Different Parts of Speech** — specifies whether to identify the parts of speech of parsed terms. If the value of this property is **`Yes`**, then same terms with different parts of speech are treated as different terms.

- **Ignore Parts of Speech** — accesses a window in which you can select one or more parts of speech to ignore when parsing. Click the ellipsis button to open the Ignore Parts of Speech window. Terms with the selected parts of speech are not parsed and do not appear in node results. This property is not used if the value of **Different Parts of Speech** is **`No`**.

- **Noun Groups** — specifies whether to identify noun groups. If stemming is turned on, then noun group elements are also stemmed.

- **Synonyms** — specifies a SAS data set that contains synonyms to be treated as equivalent. Default data sets are provided for several languages. You can edit these data sets or create your own. Click the ellipsis button to open the Select a SAS Table window.

- **Find Entities** — specifies whether to identify the entities of parsed terms.

- **Types of Entities** — accesses a window in which you can select one or more entity classifications to parse. Click the ellipsis button to open the Select Entity Types window.

  Note: If the value of **Find Entities** is **`Yes`** but you do not have any entity types selected in the Select Entity Types window, then all entity types are parsed. In other words, deselecting all entity types has the same effect as selecting all of them.

Transform Properties

- **Compute SVD** — specifies whether to compute the singular-value decomposition (SVD) of the term-by-document frequency matrix.

- **SVD Resolution** — specifies the resolution to use to generate the SVD dimensions.

- **Max SVD Dimensions** — specifies the maximum number, greater than or equal to 2, of SVD dimensions to generate.

- **Scale SVD Dimensions** — specifies whether to scale the SVD dimensions by the inverse of the singular values in order to generate equal variances.

- **Frequency Weighting** — specifies the frequency weighting method to use.

- **Term Weight** — specifies the term weighting method to use.

- **Roll up Terms** — specifies whether to first sort parsed terms in descending order of the value of the term weight multiplied by the square root of the number of documents and then roll up these selected terms as variables on the Document data set.

- **No. of Rolled-up Terms** — specifies the number of terms to use to roll up lower-weighted terms. This property is used if the value of **Roll up Terms** is `Yes`.

- **Drop Other Terms** — specifies whether to drop terms if they are not rolled up on the Document data set. This property is used if the value of **Roll up Terms** is `Yes`.

Cluster Properties

- **Automatically Cluster** — specifies whether to perform clustering analysis. If you select `No`, then the remaining Cluster Properties are not used.

- **Exact or Maximum Number** — specifies whether to find an exact number of clusters or any number less than or equal to a maximum number of clusters.

- **Number of Clusters** — specifies the number of clusters; this is the exact number if the value of **Exact or Maximum Number** is `Exact`, and it is the maximum number if the value of **Exact or Maximum Number** is `Maximum`.

- **Cluster Algorithm** — specifies the clustering algorithm.

- **Ignore Outliers** — specifies whether to ignore outliers in the clustering analysis. If the value of this property is `No`, then do one of the following:
  - for Hierarchical clustering, outliers are removed
  - for Expectation-Maximization clustering, outliers are placed in a single cluster

- **Hierarchy Levels** — (for Hierarchical clustering) specifies the number of levels in the cluster hierarchy. To specify the maximum depth (all levels), enter a period (.).

- **Descriptive Terms** — specifies the number of descriptive terms to display in each cluster.

- **What to Cluster** — specifies whether to cluster roll-up terms or the SVD dimensions.

Text Miner Node Status Properties

These are the status properties that are displayed on the **Text Miner** node:

- **Create Time** — time that the node was created.
- **Run ID** — identifier of the run of the node. A new identifier is assigned every time the node is run.
- **Last Error** — error message, if any, from the last run.
- **Last Status** — last reported status of the node.
- **Last Run Time** — time at which the node was last run.
- **Run Duration** — length of time required to complete the last node run.
- **Grid Host** — grid host, if any, that was used for computation.
- **User-Added Node** — denotes whether the node was created by a user as a SAS Enterprise Miner Extension node. The value of this property is always **No** for the **Text Miner** node.

Text Miner Node Results

Overview of Text Miner Node Results

Note: The **Text Miner** node is not available from the **Text Mining** tab in SAS Text Miner 12.1. The **Text Miner** node has now been replaced by the functionality in other SAS Text Miner nodes. You can import diagrams from a previous release of SAS Text Miner that had a **Text Miner** node in the process flow diagram. However, new **Text Miner** nodes can no longer be created, and property values cannot be changed in imported **Text Miner** nodes. For more information, see "Converting SAS Text Miner Diagrams from a Previous Version" on page 11.

- "Results Window for the Text Miner Node" on page 121
- "Text Miner Node Graphical and Tabular Results" on page 122
- "Text Miner Node SAS Output Results" on page 123

Results Window for the Text Miner Node

After the **Text Miner** node runs successfully, you can access the Results window in three ways:

- Click **Results** in the Run Status window that opens immediately after a successful run of the **Text Miner** node.

- Click the Results icon on the main Toolbar.
- Right-click the **Text Miner** node, and select **Results**.

The Results window for the **Text Miner** node is similar to the Results window for other nodes in SAS Enterprise Miner. For general information about the Results window, see the SAS Enterprise Miner Help. The icons and main menus are similar to those of other nodes. However, the **View** menu for the **Text Miner** node Results window includes selections to access graphical and tabular results.

Note: You can access the SAS log that was generated by the node processing from the **SAS Results** submenu of the **View** menu. This log can be a useful debugging tool.

Text Miner Node Graphical and Tabular Results

The following are the graphical results in the **Text Miner** node Results window:

- The **Terms: Attribute by Freq** bar chart displays the total frequency of occurrence of terms in the document collection, broken down by attribute. If you position the mouse pointer over a bar, then a tooltip indicates the attribute name and the number of times a term with that attribute appears in the entire document collection.
- The **Terms: # of Docs by Frequency: Histogram** displays the total frequency of occurrence of terms in the document collection, broken down by binned number of documents. If you position the mouse pointer over a bar, then a tooltip indicates the range of the bin and the number of terms that appear in a total number of documents within that range.
- The **Terms: # of Docs by Frequency: Scatter Plot** displays the number of documents in which a term appears versus the frequency of occurrence of that term in the entire document collection. Each data point represents a term. If you position the mouse pointer over a plotted point, then a tooltip indicates the term name, the number of documents in which that term appears, and the number of times that term appears in the entire document collection.
- The **Terms: Freq by Weight** scatter plot displays the frequency of occurrence of terms in the document collection versus the weight of the term. Each data point represents a term. If you position the mouse pointer over a plotted point, then a tooltip indicates the term name, the frequency of occurrence of the term, and the weight of the term.
- The **Terms: Role by Freq** bar chart displays the total frequency of occurrence of terms in the document collection, broken down by term role. Each bar represents a role. If you position the mouse pointer over a bar, then a tooltip indicates the role name and the number of times a term with that role appears in the entire document collection.
- The **Clusters: Freq by RMS** scatter plot (if clustering was performed) displays the number of terms in the clusters versus the root mean squared standard deviation of the cluster. Each data point represents a cluster. If you position the mouse pointer over a plotted point, then a tooltip indicates the number of terms in the cluster, the root mean squared standard deviation of the cluster, and the terms that describe the cluster.

There are two tabular results in the **Text Miner** node Results window:

- The **Terms** table displays information about top-level terms (that is, terms that have no parents above them). All graphical results in the **Text Miner** node Results window are linked to this table. Therefore, you can select an observation in the Terms table, and the associated data points are highlighted in the graphics. Or you can select data points in the graphics, and the associated observations are highlighted in the Terms table.

- (If clustering was performed) The **Clusters** table displays clustering information including cluster ID, percentage of documents in a cluster, number of documents in a cluster, root mean squared standard deviation of the cluster, and terms that describe the cluster.

Table 9.1 *Contents of the Terms Table*

Variable	Description
Term	top-level terms (in lowercase); terms that are parents are preceded by a plus (+) symbol.
Role	part of speech of the term, entity classification of the term, or the value **`Noun Group`**.
Attribute	attribute of the term.
Freq	number of times the term appears in the document collection.
# Documents	number of documents in the collection in which the term appears.
Keep	**`Y`** if the term is used in the text mining analysis.
Weight	weight of the term

Text Miner Node SAS Output Results

The SAS output from the **Text Miner** node includes variable summary, summary statistics, and information about the terms. Terms in the output are sorted by their weights.

Text Miner Node Output Data

Note: The **Text Miner** node is not available from the **Text Mining** tab in SAS Text Miner 12.1. The **Text Miner** node has now been replaced by the functionality in other SAS Text Miner nodes. You can import diagrams from a previous release of SAS Text Miner that had a **Text Miner** node in the process flow diagram. However, new **Text Miner** nodes can no longer be created, and property values cannot be changed in imported **Text Miner** nodes. For more information, see “Converting SAS Text Miner Diagrams from a Previous Version” on page 11.

The **Text Miner** node outputs the Documents, Validate, Test, Terms, Cluster, and Out data sets. The Validate and Test data sets exist only if the process flow diagram has a **Data Partition** node before the **Text Miner** node. The Cluster data set exists only if you have clustered the documents. To view the output data sets, click the ellipsis that are associated with the **Exported Data** property of the **Text Miner** node after the node has been run.

The following table lists the variables in the **Documents**, **Validate**, and **Test** data sets:

Variable	Role
original variables from the input data sources	the roles that are defined in the input data sources
Document ID (_document_)	ID
SVD dimensions (_SVD_<num>)	Input for the dimensions that are used in a cluster analysis. Otherwise, the role is set to Rejected.
the length of the vector formed from the SVD dimensions that are used (_SVDLEN_)	Rejected
Cluster ID (_CLUSTER_)	Segment
the probability that a document is categorized into a cluster if expectation-maximization clustering is performed (PROB <num>)	Prediction

The following table lists the variables and their roles in the **Terms** data set.

Variable	Role
Term	Text
Role	Input
Attribute	Input
Freq	Frequency
# Documents	Input
Keep	Input
Term ID	ID
Parent ID	Input
Parent_id	ID
_ispar	Input
oldkey	Input
Weight	Input

Variable	Role
SVD dimensions (_SVD_ <num>)	Input for the dimensions that are used in a cluster analysis. Otherwise, the role is set to Rejected.
the length of the vector formed from the SVD dimensions that are used (_SVDLEN_)	Rejected

The following table lists the variables and their roles in the **Cluster** data source.

Variable	Role
#	ID
the percentage of documents that are in a cluster (Percentage)	Rejected
SVD dimensions (_SVD_<num>)	Input
Freq	Rejected
RMS Std.	Rejected
Descriptive Terms	Rejected

Note: The Cluster data source contains only the SVD dimensions that are used in a cluster analysis.

The **Text Miner** node creates an OUT data set to be used as an input to the **Association** node with the role Transaction and the variables _termid_, _DOCUMENT_, and _count_. The _termid_ variable has the role of Target. The OUT data set has the roles and variables required by the **Association** node. Therefore, the **Text Miner** node can be linked to the **Association** node to run a flow diagram with an input data source. For details about the **Association** node, see the SAS Enterprise Miner documentation.

The following tables list the variables and their roles in the **OUT** data set.

Variable	Role
termid	Target
DOCUMENT	ID
count	Rejected

Chapter 10
Output Data

Output Data for SAS Text Miner Nodes

You can access the output data from a SAS Text Miner node by using data sets that are referenced by the macro variables &EM_DATA_EMINFO and &EM_IMPORT_DATA_EMINFO. When you run a process flow diagram in SAS Enterprise Miner, a data set is created and passed on in the flow. This data set, when exported by a node, is referenced by the macro variable &EM_DATA_EMINFO and, when imported by a node (from a previous node), is referenced by the macro variable &EM_IMPORT_DATA_EMINFO.

The EMINFO data sets keep track of which node produced the most recent output. After training is performed, this data set is edited as follows:

- Observations are added (if not already present) for the variable **`key`** and given the following values, which are discussed in detail below:
  - **`LastTMNode`**
  - **`PRESCORECODE`**
  - (if Text Parsing node) **`LastTextParsing`**
  - (if Text Filter node) **`LastTextFilter`**
  - (if Text Topic node) **`LastTopic`**
  - (if Text Cluster node) **`LastCluster`**
  - (if Text Rule Builder node) **`LastTextRule`**
- For each observation, the variable **`data`** is the assigned value of the node ID of a particular node in the process flow.
- The variable **`target`** is in the data set, but is not used by the SAS Text Miner nodes.

The value of data that corresponds to LastTextParsing, LastTextFilter, LastTopic, and LastCluster is the node ID of the most recently run node of each node type in the flow.

You could use this value, for example, to access output data sets from the most recently run node of a particular type. For example, the following code would obtain the ID of the most recently run Text Topic node run and print the TERMS output data from that node.

```
%let last_topic_node = ;
proc sql noprint;
  select data into :last_topic_node
  from &EM_IMPORT_DATA_EMINFO
  where key="LastTopic";
quit;

proc print data=&last_topic_node._TERMS
run;
```

Note: SAS Text Miner nodes use the naming convention “nodeID._TERMS” for the Terms data set.

The EMINFO data set also includes a key for PRESCORECODE. Its data value is the node ID of the last SAS Text Miner node in the process flow. SAS Enterprise Miner looks for a file called PRESCORECODE.sas in the workspace directory for the referenced node (&EM_NODEDIR.&EM_DSEP.PRESCORECODE.sas). This file contains SAS code that is appended to a score flow before it runs.

Or, if you wanted to create a SAS code node that mimicked a Text Filter node, you could add the following code to the end of your code:

```
data &EM_DATA_EMINFO;
  length TARGET KEY $32 DATA $43;

  key="LastTMNode";
  data="&EM_NODEID";
  output;

  key="LastTextFilter";
  data="&EM_NODEID";
  output;
run;
```

Chapter 11

Macro Variables, Macros, and Functions

Using Macro Variables to Set Additional Properties

Contents

- "Overview of SAS Text Miner Macro Variables" on page 130
- "Text Filter Node Macro Variables" on page 130
- "Text Topic Node Macro Variables" on page 131
- "Text Miner Node Macro Variables" on page 132
- "Text Rule Builder Node Macro Variables" on page 133

Overview of SAS Text Miner Macro Variables

Some advanced properties can be set using the value of macro variables (rather than in the node properties) for the **Text Filter** node, the **Text Topic** node, and the **Text Rule Builder** node. You can set the values of these macro variables in the start-up code. If you specify an invalid value for a macro variable, then the default value is used.

Text Filter Node Macro Variables

You can use the macro variables listed in the following table to set additional properties for the **Text Filter** node. These properties all relate to spell-checking. Note the following:

- To reduce the number of falsely identified misspellings (Type I error rate), set larger values of TMM_DICTPEN, TMM_MINPARENT, and TMM_MULTIPEN and smaller values of TMM_MAXCHILD and TMM_MAXSPEDIS.

- To reduce the number of non-identified misspellings (Type II error rate), set larger values of TMM_MAXCHILD and TMM_MAXSPEDIS and smaller values of TMM_DICTPEN, TMM_MINPARENT, and TMM_MULTIPEN.

Table 11.1 *Macro Variables for the Text Filter Node*

Macro Variable	Default Value	Description
TMM_DICTPEN	2	penalty to apply to the SPEDIS() value if a dictionary data set is specified and a potential misspelling exists in the dictionary
TMM_MAXCHILD	log10(numdocs)+1	maximum number of documents in which a term can occur and also be considered a misspelling
TMM_MAXSPEDIS	15	maximum SPEDIS() for a misspelling and its parent to have and also evaluate as a misspelling
TMM_MINPARENT	log10(numdocs)+4	minimum number of documents in which a term must occur to be considered a possible parent
TMM_MULTIPEN	2	penalty to apply to the SPEDIS() value if one of the terms is a multi-word term

Text Topic Node Macro Variables

You can use the macro variables listed in the following table to set additional properties for the **Text Topic** node.

Note: To reduce the number of falsely identified topics and to increase the number of non-identified topics, set larger values of TMM_TERM_CUTOFF_PCT.

Table 11.2 *Macro Variables for the Text Topic Node*

Macro Variable	Default Value	Description
TMM_DOCCUTOFF	0.03	document cutoff value is for any topic. It is used to determine the default document cutoff for user topics and multi-term topics in the Topic table. Higher values decrease the number of documents assigned to a topic. The document cutoff value is multiplied by 2 for multi-term topics due to the disparate number of terms in multi-term topics. For example, for a default of .03, user topics have a document cutoff of .03 and multi-term topics have a document cutoff of .06.
TMM_MAX_TOPIC_ ANGLE	3	maximum cosine allowed between two topics for them to be considered equivalent; a lower number eliminates fewer topics based on closeness to other topics; a higher number eliminates more topics based on closeness to other topics. For example, to change the maximum topic angle for exclusion of a topic to be 5 degrees, put “%let tmm_max_topic_angle=5;” in your project start-up code. A single-term topic or multi-term topic is excluded if its topic vector is within this many degrees of any lower number topic created.

Macro Variable	Default Value	Description
TMM_NORM_PIVOT	0.5	value (between 0 and 1) used for the pivot normalization of document length. If you want longer documents to contain many more topics than short documents, set this value closer to 1. If you want short documents and long documents to contain about the same number of topics, set this value closer to 0.
TMM_TERM_CUTOFF_PCT	0.1	for any multi-term topic, the initial term cutoff set to this value multiplied by the highest weighted term. For example, if the highest weighted term is 2, the default term cutoff for that multi-term topic will be .2. For multi-term topics, any terms where the absolute value of the weight is lower than this number times the maximum weight are not included, by default.

Text Miner Node Macro Variables

Note: The **Text Miner** node is not available from the **Text Mining** tab in SAS Text Miner 12.1. The **Text Miner** node has now been replaced by the functionality in other SAS Text Miner nodes. You can import diagrams from a previous release of SAS Text Miner that had a **Text Miner** node in the process flow diagram. However, new **Text Miner** nodes can no longer be created, and property values cannot be changed in imported **Text Miner** nodes. For more information, see "Converting Diagrams and Session Encoding" on page 11.

The **Text Miner** node has the following macro variables:

Table 11.3 *Macro Variables for the Text Miner Node*

Macro Variable	Default Value	Description
TM_DEBUG	0	specifies (if 0) to delete some intermediate data sets or (if 1) not to delete them
TM_MINDESCTERMS	2	minimum number (an integer) of times that a term must appear in a cluster to be considered a descriptive term

Macro Variable	Default Value	Description
TM_ROLLWEIGHT	3	defines how terms are rolled up: • if `1`, roll-up terms are those that have the highest weight • if `2`, roll-up terms are those that have the highest value of weight multiplied by log(numdocs+1), where “numdocs” is the number of documents in which the term occurs • if `3`, roll-up terms are those that have the highest value of weight multiplied by sqrt(numdocs) • if `4`, roll-up terms are those that have the highest value of weight multiplied by numdocs
TM_SVDDATA	0	specifies (if `0`) to delete the SVD input matrix and weights or (if `1`) not to delete them

Text Rule Builder Node Macro Variables

You can use the macro variable that is listed in the following table to set additional properties for the **Text Rule Builder** node.

Table 11.4 *Macro Variables for the Text Rule Builder Node*

Macro Variable	Default Value	Description
TMB_MAXTEXTLEN	200 bytes	identifies the length of the text variable in the Change Target Values table. Any text variable that matches on the first &tmb_maxtextlen bytes in that table is assumed to be the same document.

Overview of Macros and Functions

You can use the following SAS Text Miner macros and functions to create a synonym data set, filter a collection of documents, generate a compressed, term-by-document frequency summary data set, and other tasks:

- "%TEXTSYN Macro" on page 134
- "%TMFILTER Macro" on page 138
- "TGSCORE Function" on page 147

%TEXTSYN Macro

Contents

%TEXTSYN Macro Syntax

The %TEXTSYN macro is provided with SAS Text Miner. You can use this macro after a **Text Parsing** node has been successfully run to find and correct misspellings that appear in the input data source. It is not supported for use with the Chinese language.

The macro creates a synonym data set, which you can use in SAS Text Miner, that contains misspelled terms and candidate parents (correctly spelled terms). The data set includes the variables "term," "parent," and "category." Using optional arguments, you can also specify that the synonym data set include example usages (from up to two documents) of the misspelled terms.

Terms are selected as follows:

1. Candidate child terms that occur in a specified maximum number of documents are selected from all terms. Likewise, candidate parent terms that occur in a specified minimum number of documents are selected from all terms.

2. All combinations of the candidate parent and child terms are found where the first alphabetic character is the same for a parent and a child. The SPEDIS function (a Base SAS function) is used to calculate a distance measure in both directions for each parent-child combination. The minimum of the two distances is divided by the length of the shorter term, which defines a new distance measure. For multi-word terms and terms that do not appear in a dictionary, this distance measure is multiplied by a penalty constant.

3. Parent-child combinations with the lowest final distance measures (below a specified threshold) are output to the synonym data set.

```
%TEXTSYN (TERMDS=<libref.>SAS-data-set, SYNDS=<libref.>output-data-set,
          <, DOCDS=<libref.>SAS-data-set, OUTDS=<libref.>SAS-data-set,
          TEXTVAR=text-variable, DICT=<libref.>SAS-data-set, MNPARDOC=n,
          MXCHDDOC=n, MAXSPED=n, MULTIPEN=n, DICTPEN=n, CONTEXT=n >)
```

Required Arguments:

- **TERMDS=**<libref.>SAS-data-set

  specifies the name of the Terms data set that was output from any **Text Parsing** node.

  Note: Each **Text Parsing** node outputs a terms data set, which is stored with the project data. The name of these data sets follows the format NodeID_TERMS (for example, TEXTparsing_TERMS).

- **SYNDS=**<libref.>SAS-data-set

  specifies the name to give the output synonym data set. It is recommended that you save this data set in a permanent library. For more information, see "%TEXTSYN Macro Output Data Set" on page 136.

Optional arguments:

- **DOCDS=**<libref.>SAS-data-set

  specifies the name of the Documents data set that was output. This argument is required to include example usages of the misspelled terms.

- **OUTDS=**<libref.>SAS-data-set

  specifies the name of the OUT data set that was output from a **Text Parsing** node. This argument is required to include example usages of the misspelled terms.

- **TEXTVAR=**variable-name

  specifies the name of the character variable in the input data set that contains the text (or a path to a file that contains the text) to score. This argument is required to include example usages of the misspelled terms.

- **DICT=**<libref.>SAS-data-set

  specifies the name of a dictionary data set. Using a dictionary data set can reduce the number of false positives that are in the output synonym data set. For more information, see "How to Create a Dictionary Data Set" on page 56.

- **MNPARDOC=**n

  specifies the minimum number of documents in which a term must appear in order to be considered a parent term. The default value is 3.

- **MXCHDDOC=**n

  specifies the maximum number of documents in which a term must appear in order to be considered a child term. The default value is 6.

- **MAXSPED=**n

  specifies the SPEDIS() cutoff value for acceptance that a parent-child combination is valid. The default value is 15.

- **MULTIPEN=**n

  specifies the number by which to multiply the value of MAXSPED= when the parent or child term is a multi-word term. The default value is 2.

- **DICTPEN=**n

  specifies the number by which to multiply the value of MAXSPED= when the child term is found in the dictionary table. The default value is 2.

- **CONTEXT=**n

  specifies the number of words to include before and after the target term in examples. The default value is 4.

%TEXTSYN Macro Output Data Set

The synonym output data set contains the following variables:

Column Name	Description
<Example1>	(if DOCDS=, OUTDS=, and TEXTVAR= arguments are specified) example from a document that includes the TERM.
<Example2>	(if DOCDS=, OUTDS=, and TEXTVAR= arguments are specified) example from a document that includes the TERM.
TERM	child term from the KEY data set; this is typically a misspelled word.
PARENT	parent term for the misspelled word.
CATEGORY	(if part-of-speech or entity tagging is on) part-of-speech or entity of TERM
CHILDNDOCS	number of documents that contain TERM
NUMDOCS	number of documents that contain PARENT
MINSPED	minimum spelling distance between terms; the smaller the number, the closer the terms.
DICT	(if DICT= argument is specified) **Y** if TERM is found in the dictionary data set, and **N** otherwise.

%TEXTSYN Macro Example

This example assumes that you have already created a diagram in a SAS Enterprise Miner project and that you have created an input data source using the SAMPSIO.NEWS sample data set.

1. Drag the input data source that you have created using the SAMPSIO.NEWS data set into the diagram workspace.

2. Select the **Text Mining** tab on the Toolbar.

3. Drag a **Text Parsing** node into the diagram workspace.

4. Connect the input data source node to the **Text Parsing** node.

5. Select the **Text Parsing** node to view its properties.

   - Set **Different Parts of Speech** to **`No`**.

   - Set **Noun Groups** to **`No`**.

6. In the diagram workspace, right-click the **Text Parsing** node and select **Run**. Click **Yes** in the Confirmation dialog box.

7. Click **OK** in the Run Status dialog box.

8. Drag a **SAS Code** node into the diagram workspace from the **Utility** tab on the toolbar.

9. Connect the **Text Parsing** node to the **SAS Code** node.

10. Select the **SAS Code** node, and then click the ellipsis button for the **Code Editor** property.

11. Enter the following code into the code window. Be sure to change <libref> to the actual libref for your diagram.

    TIP To determine the libref of a diagram, select the diagram name in the Project panel and view the value of the **ID** property. This value is the libref.

    Note: The following code assumes that the “Mylib” libref has been assigned to a SAS library. If you have not already created a “Mylib” libref, change

    ```
    Mylib
    ```

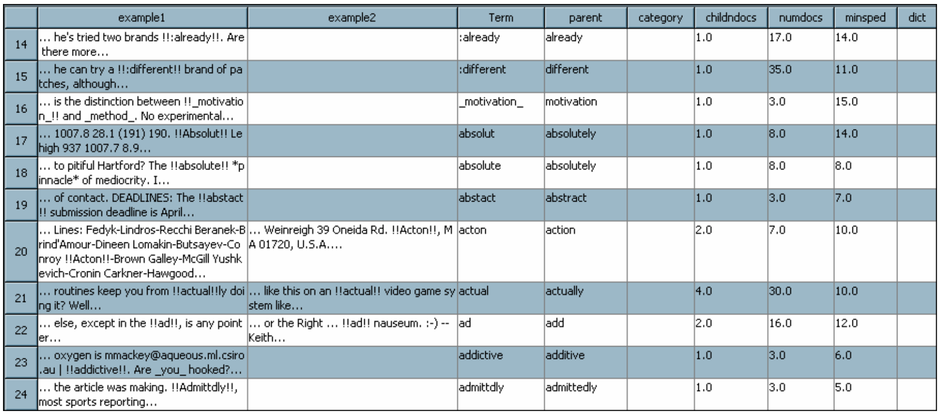
    in the following code to another libref to store your output.

    ```
    %textsyn(docds=<libref>.TEXTparsing_train,
             termds= <libref>.TEXTparsing_TERMS,
             outds=<libref>.TEXTparsing_tmOUT,
             synds=Mylib.textsyn,
             textvar=text);
    ```

12. Right-click the **SAS Code** node and select **Run**. Select **Yes** in the Confirmation dialog box, and **OK** in the Run Status dialog box when the node has finished running.

13. Click the **Explorer** toolbar shortcut to open the Explorer window. In the SAS Libraries tree, navigate to the library where you stored your output results. Highlight this directory, and then double-click the file in the right panel of the Explorer window. A preview of the data set opens.

    Note: If you do not see the data set, you might need to refresh your Explorer window. Click the **Update** toolbar button to refresh your view.



	example1	example2	Term	parent	category	childndocs	numdocs	minsped	dict
14	... he's tried two brands !!:already!!. Are there more...		:already	already		1.0	17.0	14.0	
15	... he can try a !!:different!! brand of patches, although...		:different	different		1.0	35.0	11.0	
16	... is the distinction between !!_motivation_!! and _method_. No experimental...		_motivation_	motivation		1.0	3.0	15.0	
17	... 1007.8 28.1 (191) 190. !!Absolut!! Lehigh 937 1007.7 8.9...		absolut	absolutely		1.0	8.0	14.0	
18	... to pitiful Hartford? The !!absolute!! *pinnacle* of mediocrity. I...		absolute	absolutely		1.0	8.0	8.0	
19	... of contact. DEADLINES: The !!abstact!! submission deadline is April...		abstact	abstract		1.0	3.0	7.0	
20	... Lines: Fedyk-Lindros-Recchi Beranek-Brind'Amour-Dineen Lomakin-Butsayev-Conroy !!Acton!!-Brown Galley-McGill Yushkevich-Cronin Carkner-Hawgood...	... Weinreigh 39 Oneida Rd. !!Acton!!, MA 01720, U.S.A....	acton	action		2.0	7.0	10.0	
21	... routines keep you from !!actual!!ly doing it? Well...	... like this on an !!actual!! video game system like...	actual	actually		4.0	30.0	10.0	
22	... else, except in the !!ad!!, is any pointer...	... or the Right ... !!ad!! nauseum. :-) -- Keith...	ad	add		2.0	16.0	12.0	
23	... oxygen is mmackey@aqueous.ml.csiro.au | !!addictive!!. Are _you_ hooked?...		addictive	additive		1.0	3.0	6.0	
24	... the article was making. !!Admittdly!!, most sports reporting...		admittdly	admittedly		1.0	3.0	5.0	

The example1 and example2 variables provide example usages of the candidate misspelled terms from the input data set. Misspelled terms in the examples are enclosed in exclamation marks, such as **`!!:already!!`** and **`!!abstact!!`**. Proposed correctly spelled terms are contained in the parent variable.

Note: After you run the %TEXTSYN macro, you should always examine the output data set and delete any proposed misspelled terms that should remain spelled as they are in the input data set.

%TMFILTER Macro

Contents

%TMFILTER Macro Syntax

The %TMFILTER macro is provided with SAS Text Miner. It is supported in all operating systems for filtering and on Windows for crawling. The %TMFILTER macro relies on the SAS Document Conversion Server installed and running on a Windows machine. See SAS Document Conversion server for more information.

You can use this macro to do the following:

- filter a collection of documents that are saved in any supported file format and output a SAS data set that can be used to create a SAS Text Miner data source.
- Web crawl and output a SAS data set that can be used to create a SAS Text Miner data source. Web crawling retrieves the text of a starting Web page, extracts the URL links within that page, and then repeats the process within the linked pages recursively. You can restrict a crawl to the domain of the starting URL, or you can let a crawl process any linked pages that are not in the domain of the starting URL. The crawl continues until a specified number of levels of drill-down is reached or

until all the Web pages that satisfy the domain constraint are found. Web crawling is supported only on Windows operating systems.

- identify the languages of all documents in a collection.

```
%TMFILTER (DIR=path, URL=path <,DATASET=<libref.>output-data-set
           <,DEPTH=n,  DESTDIR=path, EXT=extension1 <extension2 extension3...>,
           FORCE=anything, HOST=name | IP address, LANGUAGE=ALL | language1
           <language2 language3...>, NORESTRICT=anything,  NUMBYTES=n,
           PASSWORD=password, USERNAME=username>)
```

Note:

- If you run %TMFILTER in a UTF-8 SAS session, then the macro attempts to transcode all filtered text to UTF-8 encoding so that the resulting data set can be used in a UTF-8 SAS session. In all other SAS session encodings, %TMFILTER does not transcode the data but instead assumes that the input data is in the same encoding as the SAS session. For more information, see "SAS Text Miner and SAS Session Encoding" on page 11.

- The %TMFILTER macro sets the macro variable EMEXCEPTIONSTRING to `1` if an error occurs. You can use the value of this variable to debug programs (for example, in extension nodes or SAS Code nodes).

- If you run the Web crawling aspect of %TMFILTER macro within a SAS Enterprise Miner client installation that is connected to a Windows server, then you must issue the XCMD SAS system option on the server. For details, see the post-installation instructions that are generated during the installation of SAS Enterprise Miner and SAS Text Miner.

Required arguments:

- **DIR**=path

  specifies the path to the directory that contains the documents to process. Any subfolders in this directory are processed recursively. The path might be a UNC (Universal Naming Convention) formatted network path.

- **URL**=URL

  (Web crawling only) specifies the URL—less than or equal to 255 characters in length—of the starting Web page. The value can be either in the form **`http://www.sas.com`** or **`www.sas.com`**. Web pages that are retrieved are placed in the directory that is specified by the DIR= argument. Web crawling is supported only on Windows operating systems.

  Note:

  - If the URL contains an ampersand (&), then the ampersand will be misinterpreted as a macro trigger. In this case, you must use the %NRSTR function when you specify the URL.

  - Be aware of the terms and service policy of the Web pages that you want to crawl. Some policies might restrict automatic access, and some might restrict how you use any retrieved content. Furthermore, be aware that when crawling Web pages, the %TMFILTER macro might place a high demand on a Web host.

Optional arguments:

- **DATASET=**<libref.>output-data-set

specifies the name to give the output data set. For more information, see "%TMFILTER Macro Output Data Set" on page 141. It is recommended that you save this data set in a permanent library. If you do not specify this argument, then the name of the output data set is Work.Data.

- **DEPTH=**n

  (for Web crawling only) specifies the number of levels for Web crawling. If you do not specify this argument, then the macro visits all links on the starting Web page and all links on the linked pages (in other words, `DEPTH=2)`.

- **DESTDIR=**path

  specifies the path to the directory in which to save the output plain text files. Do not make the value of DESTDIR= a subdirectory of the DIR= directory. If you do not specify this argument, then filtered plain text files that correspond to the files in DIR= directory are not produced.

  Note: If you specify a network path, then the folders in the path must already exist on the network machine. Otherwise, the macro will fail. If you specify a local path, then it is not necessary for the folders to exist; the %TMFILTER macro creates them automatically if they do not already exist.

- **EXT=**extension1 <extension2 extension3...>

  specifies one or more file types to process. Only files (with a listed extension) that are in the directory specified by the DIR= argument are processed. To specify documents with no extension, use a single period (`EXT= .`). If you do not specify this argument, then all applicable file types are processed.

- **FORCE=**anything

  specifies not to terminate the %TMFILTER macro if the directory specified by the DESTDIR= argument is not empty when the macro begins processing. Any value (for example, `1` or `'y'`) keeps the macro from terminating if the destination directory is not empty. Otherwise, if you do not specify this option or if you do not specify a value for it, the macro terminates if the destination directory is not empty.

- **HOST=**name | IP address

  specifies the name or IP address of the machine on which to run the %TMFILTER macro. If you do not specify this argument, then the macro assumes that the SAS Document Conversion Server will use its own defaults.

- **LANGUAGE=**ALL | language1 <language2 language3...>

  specifies one or more licensed languages in which the input documents are written. If a list is supplied, then the %TMFILTER macro automatically detects the language (from those in the list) of each document. To search over all supported languages, specify `LANGUAGE=ALL`. Automatic detection is not accurate in documents that contain fewer than 256 characters.

  If you do not specify this argument, then it is assumed that input files are written in English.

- **NORESTRICT=**anything

  (for Web crawling only) specifies to allow the processing of Web sites outside the domain of the starting URL. Any value (for example, `1` or `'y'`) lets the macro process Web sites outside of the starting domain. Otherwise, if you do not specify this option or if you do not specify a value for it, only Web pages that are in the same domain as the starting URL are processed.

- **NUMBYTES**=n

  specifies the length of the TEXT variable in the output data set, in bytes. If you do not specify this argument, then the TEXT variable is restricted to 60 bytes. For more information, see "%TMFILTER Macro Output Data Set" on page 141.

- **PASSWORD**=password

  (for Web crawling only) specifies the password to use when the URL input refers to a secured Web site that requires a password.

- **PORT**=port-number

  specifies the number of the port on which the SAS Document Conversion Server resides. If it is not set, it is assumed that the conversion server will use its own defaults.

- **USERNAME**=user name

  (for Web crawling only) specifies the user name to use when the URL input refers to a secured Web site that requires a user name.

%TMFILTER Macro Output Data Set

The output data set contains the following variables:

Column Name	Description
TEXT	text of each document, truncated to the length specified by the NUMBYTES= argument.
URI	path to the input files that reside in the directory specified by the DIR= option.
NAME	name of the input file
FILTERED	path to the directory that contains the HTML file. This path corresponds to the value of the DESTDIR= argument.
LANGUAGE	most likely source language of the document, as determined by the LANGUAGE= option.
CREATED	date and time that the input document was created.
ACCESSED	date and time that the input document was last accessed.
MODIFIED	date and time that the input document was last modified.

Column Name	Description
TRUNCATED	`1` if TEXT contains truncated text; `0` otherwise.
OMITTED	`1` if the document was skipped (because of an unsupported input file type or some filtering error); `0` otherwise.
EXTENSION	file extension of the input document.
SIZE	size of the input document, in bytes.
FILTEREDSIZE	size of the input document after filtering, in bytes.

%TMFILTER Macro Details

Contents

Starting the cfs.exe Process Manually

By default, the cfs.exe process is automatically started by the %TMFILTER macro. However, if the socket connection mechanism is unable to function properly, you might need to start the process manually. To manually start cfs.exe:

1. Ensure that the log file (cfs.log) is created in a writable system location by editing the _cfsoptions.txt file. This file is located in the !SASROOT/tmine/sasmisc directory, where !SASROOT is the actual path to your SAS installation. Edit the following line, ensuring that what you specify for **`<path>`** is a writable directory, and save the file:

   ```
   <cat name="CatLogFileManager.FilePath" value="<path>\_cfsLog.txt" />
   ```

2. Open a Command Prompt window and navigate to the !SASROOT/tmine/sasmisc directory.

3. In the Command Prompt window, submit the following code. Replace <portnumber> with a four-digit port number and !SASROOT with the actual path to your SAS installation (if this path has spaces, enclose the entire path in quotation marks):

   ```
   cfs.exe -optionfile !SASROOT/tmine/sasmisc/_cfsoptions.txt -port <portnumber>
   ```

At this point, the cfs.exe process is running. The command window should no longer contain a command prompt. When you run the %TMFILTER macro, you must use the MANUAL= and PORT= options. When the macro finishes, you can either terminate the cfs.exe process (by closing the command window) or you can call the macro again without restarting the process.

Supported Document Formats

The following four tables list the proprietary-encoded file formats supported by the %TMFILTER macro.

Table 11.5 *Text, Markup, and Word Processing Formats*

Document Format	Version
ASCII text	All
ANSI text	All
HTML/XML	
Microsoft Rich Text Format (RTF)	All
Microsoft Word for Macintosh (DOC)	4 — 6, 98, 2004
Microsoft Word for PC (DOC)	4, 5, 5.5, 6
Microsoft Word for Windows (DOC)	1 — 2003
Microsoft Word for Windows (DCX)	
Microsoft WordPad	All
Microsoft Works (WPS)	1 — 4 (2000)
Office Writer	4.0 — 6.0
OpenOffice (SXW)	1, 1.1

Table 11.6 *Spreadsheets*

Document Format	Version
Comma-Separated Values (CSV)	All
Microsoft Excel for Macintosh	2.2 – 2003
Microsoft Excel for Windows (XLS)	3.0 — 4.0, 98 — 2004
Microsoft Excel for Windows (XLSX)	
Microsoft Multiplan	4
Microsoft Works Spreadsheet (S30, S40)	1 — 4
OpenOffice Spreadsheets (SXI, SXP)	1.1

Table 11.7 *Presentation Formats*

Document Format	Version
Adobe Portable Document Format (PDF)	2.0 — 6.0, Japanese
Microsoft PowerPoint for Windows (PPT)	95 — 2003
Microsoft PowerPoint for Macintosh (PPT)	4.0 — 2003
Microsoft PowerPoint (PPTX)	
OpenOffice (SXI, SXP)	1.1

Table 11.8 *Database Formats*

Document Format	Version
Access	Through 2.0
dBASE	Through 5.0
Microsoft Works for Windows	Through 4.0
Microsoft Works for DOS	Through 2.0
Microsoft Works for Macintosh	Through 2.0

%TMFILTER Macro Examples

Contents

Filter Documents of Different File Types

When you use the %TMFILTER macro to create a SAS data set that contains the text of documents of various formats, you must decide whether you want to store all the text in the variable in the SAS data set, or whether you want to reference the files by path and then place only a portion of text into the Text variable. In this example, all the text is placed in the SAS data set because the files are short. If the files were longer, the DESTDIR parameter would have been used in addition to the DIR and NUMBYTES parameters.

```
%tmfilter(dataset=tmlib.txtinput, dir=c:\public\example, numbytes=32000);
```

Note: In the above example, you must first specify a libref (such as "tmlib") to a location that you can access and specify a directory of files that you would like to process with the %TMFILTER macro.

You use the DATASET parameter to specify a name for the generated SAS data set. The DIR parameter specifies a directory that contains documents of various formats. The %TMFILTER macro extracts text of all the supported file formats in the directory and its subdirectories. In this example, the C:\public\example directory contains four files and a folder named “more” that contains three additional files.

After the SAS data set is generated, you can create a data source from this SAS data set and explore its contents. For information about how to explore the SAS data set, see the SAS Enterprise Miner Help.

In this example, the TEXT variable for the seventh row is missing because the corresponding file is an image file.

The following table shows examples of the values of the TRUNCATED and OMITTED variables:

Table 11.9 *Example Variable Values*

EXPLANATION	TRUNCATED	OMITTED
The document cannot be processed.	1	1
The document is larger than 32 KB and the text must be truncated.	1	0
The document has been processed.	0	0

If the %TMFILTER macro fails, consult the log file. The log file contains information that can help you diagnose problems that might occur during this process. Sometimes the macro skips a file because the file cannot be extracted. In this case, the filename is listed in the corresponding log file. The log file can be found by selecting the **Log** tab in the SAS Enterprise Miner Program Editor.

When you use the %TMFILTER macro to retrieve Web pages, you must specify values for the URL, DIR, and DESTDIR parameters. The %TMFILTER macro extracts text from the Web pages and transforms textual data of different formats into a SAS data set. That data set is then used as input for the **Text Parsing** node. Notice that non-text components such as figures and graphics are ignored in the process. The following example code shows how to use the %TMFILTER macro to process documents on a Web page:

```
%tmfilter(url=http://www.sas.com/technologies/analytics/datamining,
          depth=1,
          dir=c:\macro\dir,
          destdir=c:\macro\destdir,
          norestrict=1,
          dataset=tmlib.macrooutput);
```

In this example, the following Web page contains links to HTML files:

```
http://www.sas.com/technologies/analytics/datamining
```

View the SAS data set Tmlib.Macrooutput in SAS. The data set consists of the variables TEXT, URI, NAME, FILTERED, LANGUAGE, CREATED, ACCESSED, MODIFIIED, TRUNCATED, OMITTED, EXTENSION, SIZE, and FILTEREDSIZE.

When the SAS data set Tmlib.Macrooutput is used as an input to a **Text Parsing** node, the variables TEXT, FILTERED, and URI should be assigned the roles of Text, Text Location, and Web Address, respectively.

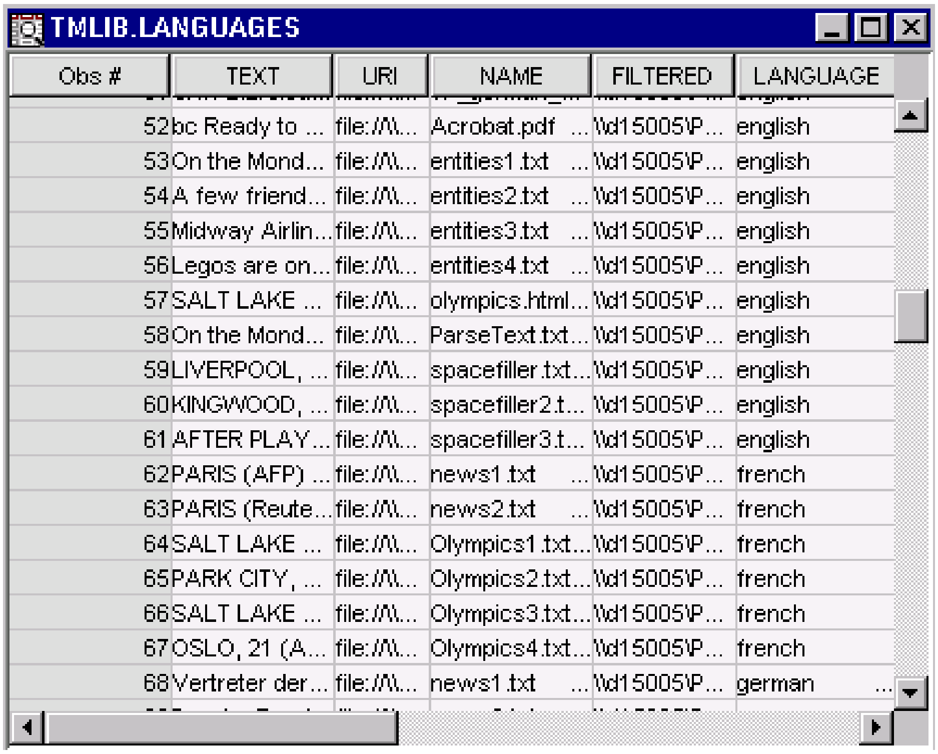
Language Detection

Suppose you store a collection of documents in various languages in the mydoc folder. You use the following %TMFILTER macro statement with the LANGUAGE parameter to create a SAS data set. The LANGUAGE parameter uses language identification technology to detect the language of each document. The value of the LANGUAGE parameter is a list of licensed languages that are separated by spaces.

```
%tmfilter(dir=c:\mydoc,
          dataset=tmlib.languages,
          language=arabic chinese czech danish
                   dutch english finnish french
                   german greek hebrew hungarian
                   indonesian italian japanese
                   korean norwegian polish portuguese
                   romanian russian slovak spanish
                   swedish thai turkish vietnamese);
```

Use any of the following parameter values: **`arabic, chinese, czech, danish, dutch, english, finnish, french, german, greek, hebrew, hungarian, indonesian, italian, japanese, korean, norwegian, polish, portuguese, romanian, russian, slovak, spanish, swedish, thai, turkish,`** and **`vietnamese`**.

The following is a display of the contents of the resulting Tmlib.Languages data set. The LANGUAGE column represents the language that SAS Text Miner detects.



Obs #	TEXT	URI	NAME	FILTERED	LANGUAGE
52	bc Ready to ...	file://\\...	Acrobat.pdf ...	\\d15005\P...	english
53	On the Mond...	file://\\...	entities1.txt ...	\\d15005\P...	english
54	A few friend...	file://\\...	entities2.txt ...	\\d15005\P...	english
55	Midway Airlin...	file://\\...	entities3.txt ...	\\d15005\P...	english
56	Legos are on...	file://\\...	entities4.txt ...	\\d15005\P...	english
57	SALT LAKE ...	file://\\...	olympics.html...	\\d15005\P...	english
58	On the Mond...	file://\\...	ParseText.txt...	\\d15005\P...	english
59	LIVERPOOL, ...	file://\\...	spacefiller.txt...	\\d15005\P...	english
60	KINGWOOD, ...	file://\\...	spacefiller2.t...	\\d15005\P...	english
61	AFTER PLAY...	file://\\...	spacefiller3.t...	\\d15005\P...	english
62	PARIS (AFP) ...	file://\\...	news1.txt ...	\\d15005\P...	french
63	PARIS (Reute...	file://\\...	news2.txt ...	\\d15005\P...	french
64	SALT LAKE ...	file://\\...	Olympics1.txt...	\\d15005\P...	french
65	PARK CITY, ...	file://\\...	Olympics2.txt...	\\d15005\P...	french
66	SALT LAKE ...	file://\\...	Olympics3.txt...	\\d15005\P...	french
67	OSLO, 21 (A...	file://\\...	Olympics4.txt...	\\d15005\P...	french
68	Vertreter der...	file://\\...	news1.txt ...	\\d15005\P...	german ...

The accuracy of language identification increases significantly if all the documents in the DIR directory are in the languages that SAS Text Miner supports.

Convert Files

You can use the %TMFILTER macro to convert files into a SAS data set that can then be used as an input data source for the **Text Parsing** node. The %TMFILTER macro runs only on Windows operating environments. The following example specifies some %TMFILTER macro parameters.

1. Create a new project and define the following LIBNAME statement in the start-up code:

   ```
   libname mylib '<path-to-your-library>';
   ```

   Note: The value **`mylib`** is the first-level SAS name that identifies the library name. You must replace <path-to-your-library> with the path specification that points to your SAS library.

2. Run the following code in the Program Editor:

   ```
   %tmfilter (dataset=mylib.tmoutds,
              dir=\\aclientmachine\Public\TM\Examples,
              destdir=\\aclientmachine\Public\TM\mydatasets,
              ext=doc txt html pdf,
              language=english spanish,
              numbytes=20);
   ```

   - The value **`mylib`** is the library name, and the value tmoutds is the name of the generated SAS data set.
   - The value **`\\aclientmachine\Public\TM\Examples`** is the path to the directory that contains the files to be processed.
   - The value **`\\aclientmachine\Public\TM\mydatasets`** is the path to the directory that will contain the filtered files.
   - The file extensions .doc, .txt, .html, and .pdf are processed by the %TMFILTER macro.
   - The languages **`english`** and **`spanish`** are by the %TMFILTER macro.
   - The length in bytes of the TEXT variable is 20.

TGSCORE Function

Contents

TGSCORE Function Syntax

The TGSCORE function is called inside a SAS DATA step and is used in SAS Text Miner for document scoring. It takes a textual variable and data sets from a SAS Text Miner training run and generates a compressed, term-by-document frequency summary data set. Child term frequencies are mapped to the parent terms. The TGSCORE function can also build a Teragram search index. A zero is returned if the function runs successfully, and a nonzero value is returned otherwise.

```
TGSCORE (text-variable, "<libref.>CONFIG-data-set", "<libref.>KEY-data-set",
          "<libref.>output-data-set", "multiterm-file-path"|0, 0|1)
```

Table 11.10 Arguments of the TGSCORE Function

Argument	Description
text-variable	name of character variable in the input data set that contains the text (or a path to a file that contains the text) to score.
CONFIG-data-set	name of the CONFIG data set. This data set contains the parsing settings that are used for scoring.
KEY-data-set	name of the KEY output data set, which contains summary information about the terms from the training data at the document collection level. The data set must contain the variables “term” and “key.” It must also contain “role” if part-of-speech tagging is used, and “parent” can be included to map child terms to parent terms. • You must sort the data set by the variable “term” and also create an index on this variable if part-of-speech tagging is used. • You must sort the data set by the variables “term” and “role” and also create an index on these variables if part-of-speech tagging is not used.
output-data-set	name to give the output data set.
multiterm-file-path	file path of the XML file that contains the list of multi-word terms. The SAS Text Miner parser ensures that the multi-word terms that are on this list are treated as a single term. If you are not using a multiterm file, then you must enter a **0** (with no quotation marks) for this argument.
0|1	**1** builds a Teragram search index; **0** does not. If an index is created, then multiple index files (with different extensions) are saved to the Work directory with the name **stgindex**. These files can be used to perform queries against the data that is being scored.

TGSCORE Function Example

1. Using SAS Text Miner, build and run a model to produce a KEY output data set and a CONFIG data set. You can determine the names of these data sets by looking at the SAS log. In this example, the data sets that were produced are named EMWS.TEXT_TERMS and EMWS.TEXT_OUT_T_CFG, respectively.

2. Sort and index the KEY data set. In this example, the training run has used part-of-speech tagging. Therefore, you can create a composite index on the term and role variables. The following code accomplishes the sorting and indexing.

```
proc sort data=EMWS.TEXT_TERMS out=work.sortKey;
   by key;
run;

proc datasets lib=work nolist;
   modify sortKey;
   index create both=(term role);
run;
```

3. Use the TGSCORE function to produce an output data set and Teragram search index files in the Work directory:

```
data _NULL_;
   set sampsio.svdtutor;
   rc=tgscore(text,"EMWS.TEXT_OUT_T_CFG","WORK.sortKey","WORK.OUT",0,1);
run;
```

Note: You can use a **Score** node after a **Text Parsing** node in a process flow diagram and examine the generated code for more examples.

Chapter 12
Additional Examples

Classifying News Articles

Contents

- "Description of the Input Data Set" on page 151
- "Creating the Process Flow Diagram" on page 152
- "Viewing Results" on page 153
- "Using SAS Code Node to Generate a Precision and Recall ROC Chart" on page 154
- "Precision and Recall ROC Chart" on page 156
- "PRCRCROC and PREREC Macros" on page 156

Description of the Input Data Set

This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

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

The SAMPSIO.NEWS data set contains 600 observations and the following variables:

- **TEXT** is a nominal variable that contains the text of the news article.

- **graphics** is a binary variable that indicates whether the document belongs to the computer graphics category (1-yes, 0-no).

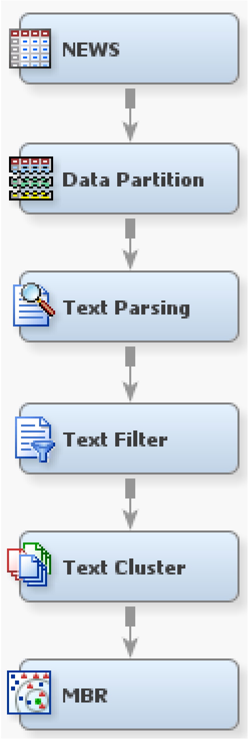
- **hockey** is a binary variable that indicates whether the document belongs to the hockey category (1-yes, 0-no).

- **medical** is a binary variable that indicates whether the document is related to medical issues (1-yes, 0-no).

- **newsgroup** is a nominal variable that contains the group that a news article fits into.

Creating the Process Flow Diagram

Follow these steps to create the Process Flow Diagram:



Input Data Source

1. Use the Data Source Wizard to define a data source for the data set SAMPSIO.NEWS.

2. Set the measurement levels of the variables graphics, hockey, and medical to `Binary`.

3. Set the model role of the variable hockey to `Target` and leave the roles of graphics, medical, and newsgroup as `Input`. The variable TEXT has a role of `Text`.

4. Select `No` in the Data Source Wizard — Decision Configuration dialog box. Use the default target profile for the target hockey.

5. After you create the data source, drag and drop it to the diagram workspace.

Data Partition Node

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

2. Set the **Partitioning Method** property to **`Simple Random`**.

3. In the Properties panel of the **Data Partition** node, change the data set percentages for training, validation, and test to 60.0, 20.0, and 20.0, respectively.

Text Parsing Node

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

2. Ensure that the **Stem Terms**, **Different Parts of Speech**, and **Noun Groups** properties are set to **`Yes`**. Use the default settings for other parsing properties.

Text Filter Node

1. Add a **Text Filter** node to the diagram workspace.

2. Connect the **Text Parsing** node to the **Text Filter** node.

Text Cluster Node

1. Add a **Text Cluster** node to the diagram workspace.

   Note: For process flow diagrams where you want to use the **Decision Tree** node in your predictive model, it is recommended that you use the **Text Topic** node instead of the **Text Cluster** node.

2. Connect the **Text Filter** node to the **Text Cluster** node.

MBR Node (Memory-Based Reasoning Node)

1. Add an **MBR** node to the diagram workspace.

2. Connect the **Text Cluster** node to the **MBR** node.

3. Right-click the **MBR** node, and select **Run**.

4. Click **Yes** in the Confirmation dialog box.

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

Viewing Results

In the MBR Results window, select **View** ⇨ **Assessment** ⇨ **Classification Chart: hockey**.

The classification chart displays the agreement between the predicted and actual target values. By default, the chart displays the percentage on the vertical axis. To create a plot based on frequency counts, right-click in the chart and select **Data Options**. In the Data Options dialog box, set the Role for the variable COUNT to **Response**.

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

Follow these steps to generate a precision and recall ROC chart:

1. Add a **SAS Code** node to the diagram workspace and connect the **MBR** node to it.

2. Select the **SAS Code** node and click the ellipsis button next to the **Code Editor** property to open the Code window.

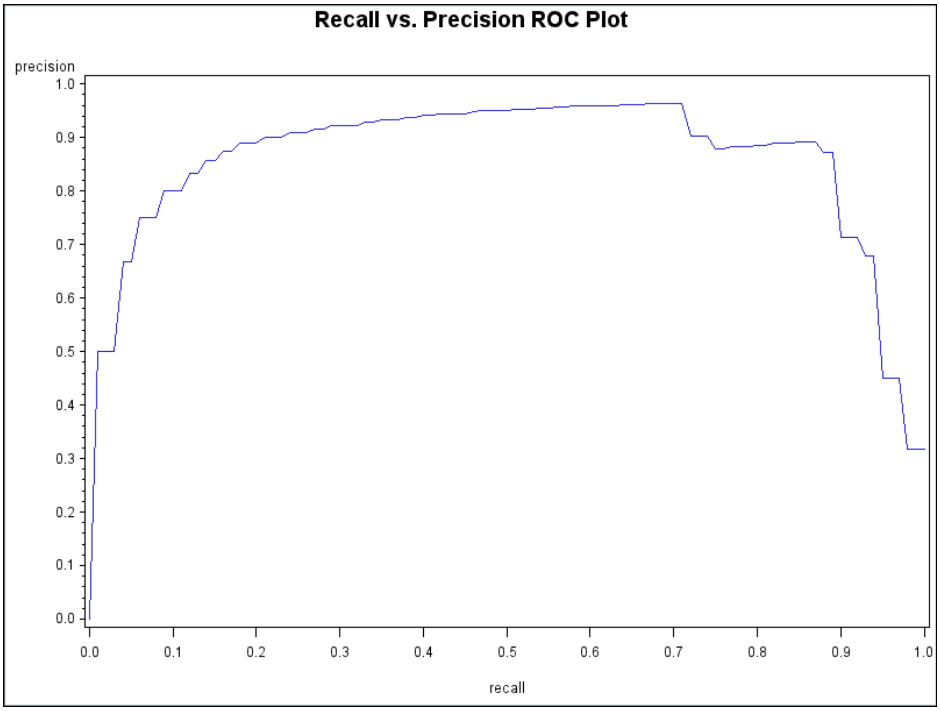
3. Enter the following SAS code in the Training Code editor:

   ```
   %prcrcroc(hockey, P_hockey1, &EM_IMPORT_TEST);
   %prerec(hockey, I_hockey, &EM_IMPORT_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 156.

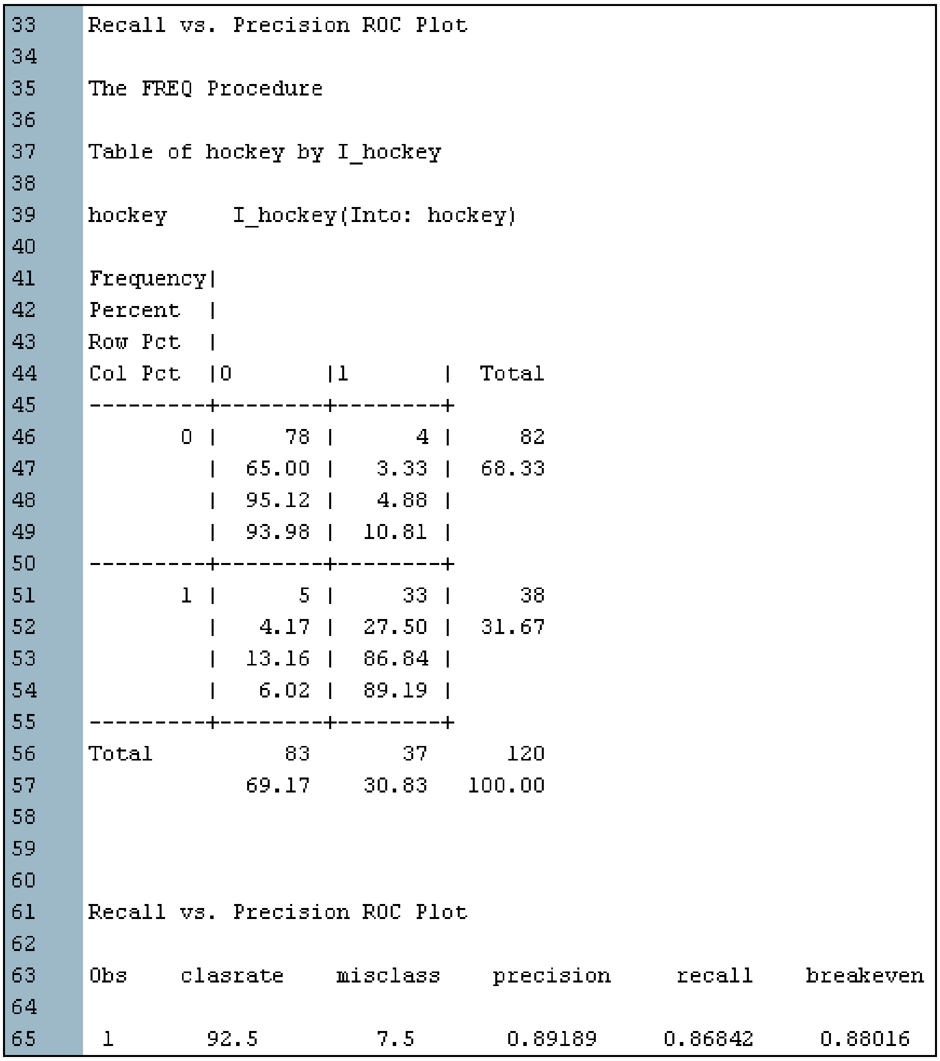
4. Close the SAS Code window. Run the **SAS Code** node and view the results.

5. In the SAS Code Results window, select from the main menu **View** ⇨ **SAS Results** ⇨ **Train Graphs** to display the precision and recall ROC chart. The following display shows the precision and recall ROC chart:



For more information, see the Precision and Recall ROC Chart topic in the SAS Enterprise Miner help.

6. The Output window in the SAS Code Results window 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 observations). The following display shows the resulting confusion matrix.



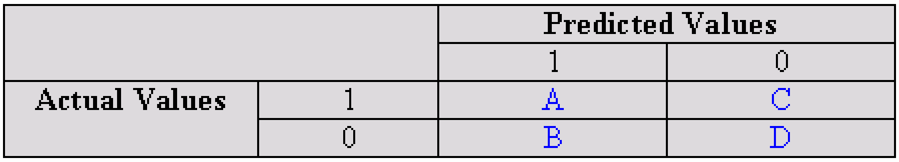
```
Recall vs. Precision ROC Plot

The FREQ Procedure

Table of hockey by I_hockey

hockey     I_hockey(Into: hockey)

Frequency|
Percent  |
Row Pct  |
Col Pct  |0       |1       |  Total
---------+--------+--------+
       0 |     78 |      4 |     82
         |  65.00 |   3.33 |  68.33
         |  95.12 |   4.88 |
         |  93.98 |  10.81 |
---------+--------+--------+
       1 |      5 |     33 |     38
         |   4.17 |  27.50 |  31.67
         |  13.16 |  86.84 |
         |   6.02 |  89.19 |
---------+--------+--------+
Total          83       37      120
            69.17    30.83   100.00



Recall vs. Precision ROC Plot

Obs    clasrate    misclass    precision     recall    breakeven

 1       92.5         7.5       0.89189     0.86842     0.88016
```

Interpretation of the matrix shows the following:

- 78 articles are correctly predicted as 0.
- 33 articles are correctly classified into the hockey category.
- 5 articles that actually belong in the hockey category are misclassified.
- 4 articles that do not belong in the hockey category are misclassified.
- The overall correct classification rate is 92.5%.
- Precision and recall are 0.89189 and 0.86842, respectively. The break-even point is 0.88016, 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 non-relevant 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, precision = A/(A+B) and recall = A/(A+C). 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.

PRCRCROC and PREREC Macros

Two macros, PRCRCROC and PREREC, are used in this example to explore the results from the **MBR** 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, &EM_IMPORT_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 &EM_IMPORT_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, I_hockey, &EM_IMPORT_TEST);
```

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

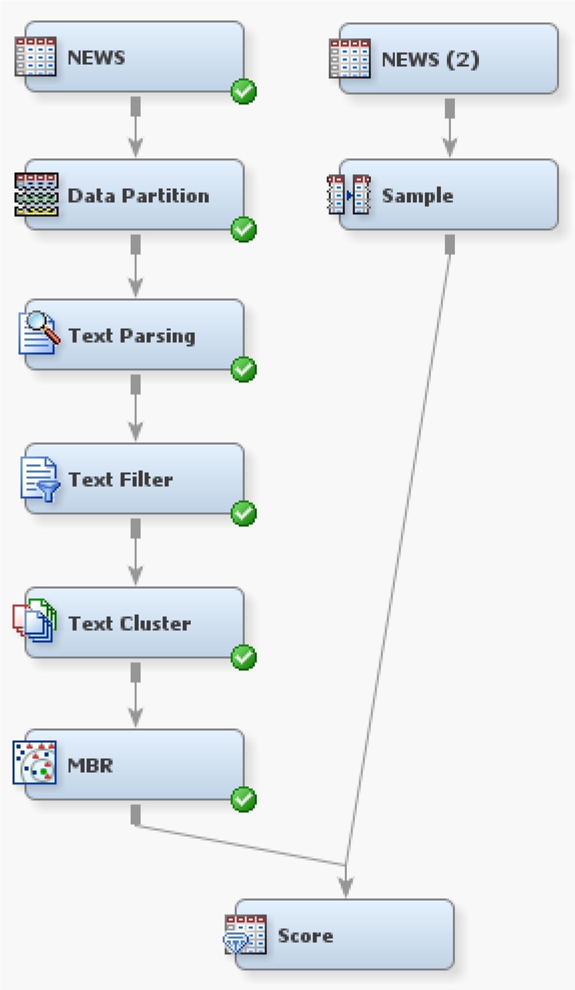
Scoring New Documents

This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

When the model for the document collection is generated, you can use it to predict 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 "Classifying News Articles" on page 151, and you want to score new documents with the model. In this case, you must use the **Score** node in your process flow diagram. 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.

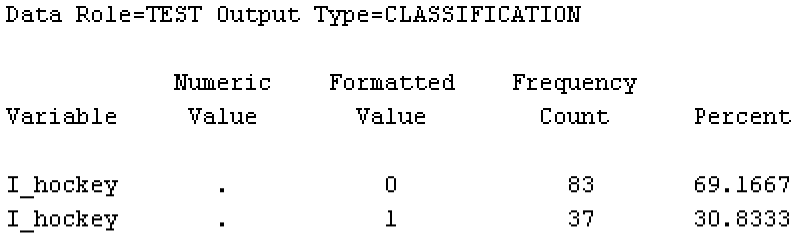
1. Create the process flow diagram in the Classification of News Articles example through the **MBR** node.
2. Drag and drop the SAMPSIO.NEWS data source onto the diagram workspace, and rename it NEWS (2).
3. Select the NEWS (2) input data node in the diagram workspace and change the value of the Role property to `Score`.
4. Select the **Sample** tab on the toolbar, and drag a **Sample** node into the diagram workspace.
5. Connect the NEWS (2) input data node to the **Sample** node.
6. Select the **Assess** tab on the toolbar, and drag a **Score** node into the diagram workspace.
7. Connect the **MBR** node to the **Score** node, and connect the **Sample** node to the **Score** node. Your process flow diagram should resemble the following:



8. Right-click the **Score** node and select **Run**. Click **Yes** in the Confirmation window that appears.

9. Click **Results** in the **Run Status** dialog box when the node finishes running. The Output window in the Results window displays the summary statistics for the score class variable, such as I_hockey.

```
Class Variable Summary Statistics

Data Role=SCORE Output Type=CLASSIFICATION

            Numeric     Formatted    Frequency
Variable     Value        Value        Count       Percent

I_hockey       .            0           39           65
I_hockey       .            1           21           35
```



```
Data Role=TEST Output Type=CLASSIFICATION

              Numeric     Formatted     Frequency
Variable       Value        Value         Count       Percent


I_hockey         .            0             83        69.1667
I_hockey         .            1             37        30.8333
```

10. Close the Results window.

Using the Association Node with SAS Text Miner Nodes

This example assumes that SAS Enterprise Miner is running, and a diagram workspace has been opened in a project. For information about creating a project and a diagram, see *Getting Started with SAS Enterprise Miner*.

You can use topic output as input to an **Association** node to help you perform association discovery and sequence discovery of topics.

Association discovery is the identification of items that occur together in a given event or record. Association discovery rules are based on frequency counts of the number of times items occur alone and in combination. An association discovery rule can be expressed as "if item A is part of an event, then item B is also part of the event X percent of the time." Associations can be written using the form A ==> B, where A (or the left hand side) is called the antecedent and B (the right hand side) is called the consequent.

Both sides of an association rule can contain more than one item. An example of an association rule might be, "If shoppers buy a jar of salsa, then they buy a bag of tortilla chips." In this example, the antecedent is "buy a jar of salsa," and the consequent is "buy a bag of tortilla chips." Association rules should not be interpreted as a direct causation. Association rules define some affinity between two or more items.

Sequence discovery takes into account the ordering of the relationships among items. For example, rule A ==> B implies that event B occurs after event A occurs. Here are two hypothetical sequence rules:

- Of those customers who currently hold an equity index fund in their portfolio, 15% of them will open an international fund in the next year.

- Of those customers who purchase a new computer, 25% of them will purchase a laser printer in the next month.

You can use the **Text Topic** node as input to an **Association** node. This example illustrates a process flow diagram that generates topic output as input for an **Association** node in SAS Enterprise Miner. Perform the following steps to use a **Text Topic** node as input to an **Association** node:

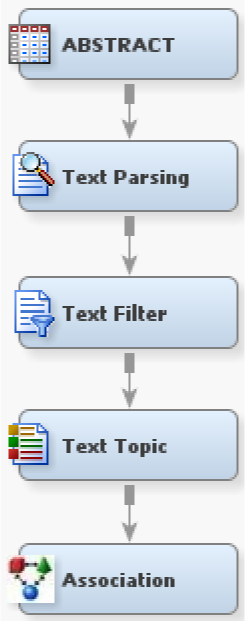
1. The SAS data set SAMPSIO.ABSTRACT contains the titles of abstracts and abstracts from conferences. Create the ABSTRACT data source and add it to your diagram workspace. Set the Role value of the TEXT and TITLE variables to `Text`.

2. Select the **Text Mining** tab on the toolbar, and drag a **Text Parsing** node into the diagram workspace. Connect the ABSTRACT data source to the **Text Parsing** node.

3. Drag a **Text Filter** node from the **Text Mining** tab into the diagram workspace. Connect the **Text Parsing** node to the **Text Filter** node.

4. Drag a **Text Topic** node from the **Text Mining** tab into the diagram workspace. Connect the **Text Filter** node to the **Text Topic** node.

5. Select the **Explore** tab on the toolbar, and drag an **Association** node into the diagram workspace. Connect the **Text Topic** node to the **Association** node. Your final diagram should resemble the following:



6. In the diagram workspace, right-click the **Association** node and select **Run**. Click **Yes** in the Confirmation dialog box that appears.

Note: Other process flow diagrams are possible. However, a **Text Topic** node should precede the **Association** node in your process flow diagram.

For more information about the **Association** node, see the SAS Enterprise Miner help.

Appendix 1
Additional Information about Text Mining

The following is recommended background reading on text mining:

- 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.
- Krishnaiah, P.R., and L.N. Kanal. 1982. *Classification, Pattern Recognition, and Reduction of Dimensionality.* New York: North-Holland Publishing Company.
- 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).

Index