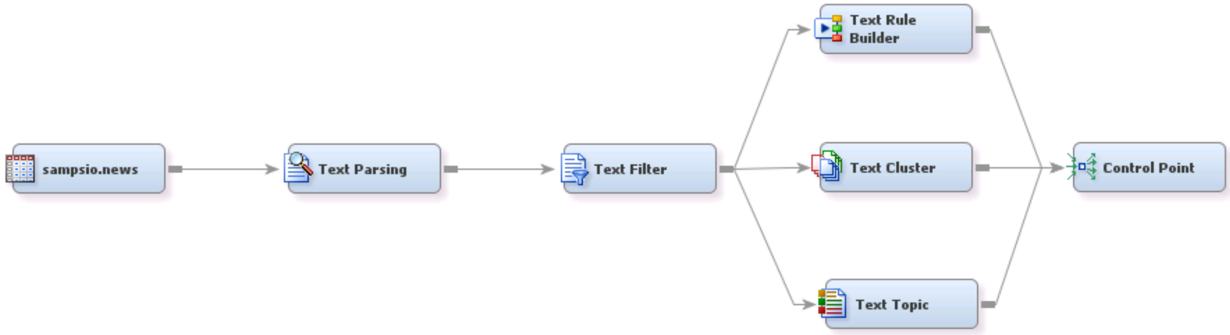
Text Mining Exploration in SAS® Enterprise Miner™



Data:

The data set for this example is SAMPSIO.NEWS, which contains 600 news articles as observations. The target is the newsgroup variable, which has three nominal outcomes: medical, hockey, and graphics. The variable TEXT has the role Text and level Nominal, and the variable newsgroup has the role Target and level Nominal. All other variables have the role Rejected because they provide no additional information.

Goal:

The goal is to use SAS Enterprise Miner text mining nodes to explore text data.

Flow:

The sampsio.news node (an Input Data node) is connected to the Text Parsing node, which is required for dealing with text data. The Text Parsing node takes the raw text from the data source and forms associations with words or word groups and articles; it can handle many languages and has a variety of properties for detecting or ignoring various parts of speech. For this example, the default settings are used.

The Text Parsing node needs to be immediately followed by a Text Filter node, which applies filters to text data. User-defined dictionaries and term weighting can be added. This example uses the default settings for weighting the log frequency and excluding terms that appear in fewer than four articles. In the **Results** window of the Text Filter node, the **Terms** table is a full list of words found in the articles, with various statistics.

The Text Parsing and Text Filter nodes create a transaction data set that associates each word (which has been given a number) to every article (document) in the input data set. For example, the word "doctor" (found in the **Terms** table as term "+ doctor" noun) is word number 13,958 and is associated with 27 articles: 410, 420, 424, and so on. The full **Terms** table can be viewed in the **Results** of the Text Filter node or by using the explorer to browse project data and looking for the table textfilter_terms (in the appropriate diagram library). Determining which articles a term is associated with requires you to view the exported transaction data from the Text Filter node.

This example discusses three nodes that can help in exploring the textual data: Text Rule Builder, Text Cluster, and Text Topic. Right-clicking the Control Point node and selecting Run causes all three parallel flows to execute.

The Text Rule Builder node constructs rules similar to those found in association mining. Default settings are used in this example, but you can change them, depending on your application. You can choose how much of the data all the rules cover and how pure the rules should be. However, setting these values too high can lead to overfitting. After you run the Text Rule Builder node, you can view the rules by clicking **Results** and looking at the **Rules Obtained** table. This table includes the Rule (often a word) and the Target Value that the rule implies. For example the word “doctor” implies that the article included the target value of “medical.”

The Text Cluster node uses SVD (singular value decomposition) to cluster the articles into multiple groups. Each article is assigned to one group, based on its components from the SVD. Default properties are used in this example, but you can change the number of dimensions for the SVD, the number of clusters, and how clustering is performed.

The Text Topic node determines textual topics in the article collection. This node differs from the Text Cluster node, whose goal is to separate the articles into disjoint groups. The Text Topic node determines the textual topics and then calculates how much each topic is represented in each article. Default settings are used for the Text Topic node in this example, but you can change the number of topics that are computed. After you run the Text Topic node, you can click the ellipses for the interactive **Topic Viewer** under the Text Topic node properties. Then you can interactively view the contents of each topic, and you can view which articles are representative of the selected topic.