Extract Word Co-Occurrence Network

⚠ This algorithm has been updated since the release of Sci2 v1.0 alpha. To use the new version of this algorithm download the edu.iu.nwb.composite.extractcowordfromtable_1.0.1.jar and copy it into the Sci2 plugins directory. Make sure to remove the old plugin, "edu.iu.nwb.composite.extractcowordfromtable_1.0.0" from the directory or the new plugin will not work. For more information on how to replace or add new plugins, see 3.2 Additional Plugins.

Description

Extract Word Co-Occurrence Network creates a weighted network where each node is a word and edges connect words to each other, where the strength of an edge represents how often two words occur in the same body of text together.

This algorithm is a shortcut for extracting a directed network using Extract Directed Network, and then performing bibliographic coupling using Extract Reference Co-Occurrence (Bibliographic Coupling) Network.

Parameters

- **Node Identifier Column** - The column whose contents identify each record, for instance if we are operating on a table of papers, you could use the title.
- **Text Source Column** -The column identifying the text which is used to determine the similarity between records. For a table of papers, this could be an abstract, for instance.
- **Text Delimiter** - The character which separates words in your Text Source Column. In most unaltered text this would be a space, but a pipe ('|') is chosen by default because it is the separator used as a result of running Lowercase, Tokenize, Stem, and Stopword Text

Implementation Details

Due to this algorithm's use of Extract Directed Network, there is an unfortunate residual effect present, where the titles of the original papers (or whatever your original records are) are present as nodes in the network. Fortunately these nodes are listed as occurring 0 times, so they can be removed by running Extract Nodes Above or Below Value, and keeping all nodes which occur more than zero times.

Usage Hints

It may be useful to run DrL (VxOrd) on the resulting network, which will lay out of the nodes in a force-directed manner, so words which are similar will be relatively close to each other.

Also, this extraction will produce edges between all words which ever occur together, which can lead to a number of edges which is difficult and costly to visualize or otherwise manipulate, so after running DrL (VxOrd), or before doing any other processing, it is often a good idea to run Extract Edges Above or Below Value, and remove all but the strongest edges, until the number of edges reaches a manageable number. You can determine the number of edges in a graph by running the Network Analysis Toolkit.

Links

- Source Code