

# Automap User's Guide 2013

**Kathleen M. Carley, Dave Columbus, Peter Landwehr**

June 03, 2013  
CMU-ISR-13-105

Institute for Software Research  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213



Center of the Computational Analysis of Social and Organization Systems (CASOS) technical report

*This report/document supercedes CMU-ISR-12-106  
"Automap User's Guide 2012", June 2012*

This work is part of the Dynamic Networks project at the center for Computational Analysis of Social and Organizational Systems (CASOS) of the School of Computer Science (SCS) at Carnegie Mellon University (CMU). This work is part of the Dynamic Networks project at the center for Computational Analysis of Social and Organizational Systems (CASOS) of the School of Computer Science (SCS) at Carnegie Mellon University (CMU). This work was supported in part by the Office of Naval Research MURI - A Structural Approach to the Incorporation of Cultural Knowledge in Adaptive Adversary Models(N00014-08-1-1186), Office of Naval Research - SORASCS - Architecture to Support Socio-Cultural Modeling (N000140811223); Office of Naval Research - CATNET: Competitive Adaptation in Terrorist Networks (N00014-09-1-0667); the Air Force Office of Sponsored Research - Multi-Level cultural Modeling - FA87500820020, and Netanomoics. Additional support was provided by CASOS - the center for Computational Analysis of Social and Organizational Systems at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, the Air Force Office of Sponsored Research, the Defense Threat Reduction Agency, the Federal Aviation Administration, Netanomics or the U.S. government.

**Keywords:** Semantic network analysis, dynamic network analysis, mental modes, social networks, AutoMap

## **Abstract**

AutoMap is an advanced text mining system. It operates in 4 modes. First, it can do classical content analysis; i.e. concepts and their frequency. Second, it extracts the semantic network; i.e. concepts and their relation to each other. Third, it cross classifies the concepts into their ontological categories such as agents and locations which results in meta-network. This includes, e.g. the social network. Fourth, it utilizes post processing to infer various aspects of sentiment.

AutoMap software is available for download from the CASOS website on its project page:  
<http://casos.cs.cmu.edu/projects/automap>



# Table of Contents

General Information .....	1
AutoMap 3 Overview .....	1
Glossary.....	3
DOS Commands .....	12
References .....	16
Data Collection.....	20
GUI Quickstart.....	21
Simple Tutorials .....	23
Content Analysis to Semantic Network .....	27
Interface Details.....	30
Script Quickstart.....	33
AM3Script Tags Details .....	37
Simple Tutorials .....	50
GUI .....	55
The GUI (Graphic User Interface).....	55
Keyboard Shortcuts.....	57
Edit Menu.....	59
Generate Menu.....	60
Generate-Parts Of Speech .....	61
Generate-Concept Lists.....	62
Generate-Semantic Networks.....	64
Generate-Meta-Networks.....	66
Generate-Thesaurus Suggestion.....	69
Generate-Generalization Thesauri.....	71
Generate - Networks Uncovered By Bayesian Inference (NUBBI).....	73
Procedures .....	81
Procedures-Master Thesauri.....	81

Procedures-Concept List .....	88
Procedures-Thesauri .....	89
Procedures-Delete Lists .....	92
Procedures-DyNetML.....	94
First Run with the GUI.....	95
Files .....	100
File Menu .....	100
File Formats.....	102
Text Encoding.....	103
Encoding Lesson.....	106
Table Viewer .....	108
Tagged Text Viewer.....	110
Concept Lists .....	115
Concept Lists .....	115
Union Concept List .....	117
Concept List Viewer .....	120
Using a Concept List .....	124
Compare Concept Lists .....	127
Thesauri.....	130
Thesauri, General.....	130
Thesauri, MetaNetwork.....	134
Thesaurus Content Only .....	136
Thesauri Editor .....	139
Compare Thesauri Files.....	143
Using a Generalization Thesaurus.....	146
Working with Large Thesauri .....	149
Semantic Lists.....	152
Semantic Lists.....	152
Other Preprocessing .....	154
Preprocessing Menu .....	154
Anaphora.....	155
Bi-Grams.....	156

Format Case .....	160
Named Entities.....	161
Parts of Speech .....	161
Remove Items .....	165
Process Sequencing.....	169
Stemming.....	170
Text Properties.....	173
Threshold, Global and Local.....	173
Union.....	178
Window Size.....	179
Networks.....	180
Networks.....	180
Semantic Networks .....	183
Meta-Network Thesaurus .....	186
XML Viewer .....	186
Extracting a Semantic Network.....	190
Scripts.....	194
AM3Script .....	194
Script Runner .....	210
First Run with the Script.....	214



## General Information

### Description

This section contains general information about AutoMap 3



## AutoMap 3 Overview

### An Overview

AutoMap is text analysis software that implements the method of Network Text Analysis, specifically Semantic Network Analysis. Semantic analysis extracts and analyzes links among words to model an author's **mental map** as a network of links. Automap also supports Content Analysis.

Coding in AutoMap is computer-assisted; the software applies a set of coding rules specified by the user in order to code the texts as networks of concepts. Coding texts as maps focuses the user on investigating meaning among texts by finding relationships among words and themes.

The coding rules in AutoMap involve text pre-processing and statement formation, which together form the coding scheme. Text pre-processing condenses data into concepts, which capture the features of the texts relevant to the user. Statement formation rules determine how to link concepts into statements.

### Network Text Analysis (NTA)

Network Text Analysis theory is based on the assumption that language and knowledge can be modeled as networks of words and relations. NTA encodes links among words to construct a network of linkages. Specifically, this method analyzes the existence, frequencies, and covariance of terms and themes, thus subsuming classical Content Analysis.

## Social Network Analysis (SNA)

Social Network Analysis (Wasserman & Faust, 1994) is a scientific area focused on the study of relations, often defined as social networks. In its basic form, a social network is a network where the nodes are people and the relations (also called links or ties) are a form of connection such as friendship. Social Network Analysis (Wasserman & Faust, 1994) takes graph theoretic ideas and applies them to the social world. The term "social network" was first coined in 1954 by J. A. Barnes (see: Class and Committees in a Norwegian Island Parish). Social network analysis (Wasserman & Faust, 1994) is also called network analysis, structural analysis, and the study of human relations. SNA is often referred to as the science of **connecting the dots**.

Today, the term Social Network Analysis (Wasserman & Faust, 1994) is used to refer to the analysis of any network such that all the nodes are of one type (e.g., all people, or all roles, or all organizations), or at most two types (e.g., people and the groups they belong to). The metrics and tools in this area, since they are based on the mathematics of graph theory, are applicable regardless of the type of nodes in the network or the reason for the connections.

For most researchers, the nodes are actors. As such, a network can be a cell of terrorists, employees of global company or simply a group of friends. However, nodes are not limited to actors. A series of computers that interact with each other or a group of interconnected libraries can also comprise a network.

## Semantic Network Analysis

In map analysis, a concept is a single idea, or ideational kernel, represented by one or more words. Concepts are equivalent to nodes in Social Network Analysis (SNA) (Wasserman & Faust, 1994). The link between two concepts is referred to as a statement, which corresponds with an edge in SNA. The relation between two concepts can differ in strength, directionality, and type. The union of all statements per texts forms a semantic map. Maps are equivalent to networks.

## Dynamic Network Analysis

Dynamic Network Analysis (DNA) is an emergent scientific field that brings together traditional social network analysis (SNA) (Wasserman & Faust,

1994), link analysis (LA) and multi-agent systems (MAS). There are two aspects of this field. The first is the statistical analysis of DNA data. The second is the utilization of simulation to address issues of network dynamics. DNA networks vary from traditional social networks in that there are larger dynamic multi-mode, multi-plex networks, and may contain varying levels of uncertainty.

DNA statistical tools are generally optimized for large-scale networks and simultaneously admit the analysis of multiple networks in which there are multiple types of entities (multi-entities) and multiple types of links (multi-plex). In contrast, SNA statistical tools focus on single or at most two mode data and facilitate the analysis of only one type of link at a time.

Because they have measures that use data drawn from multiple networks simultaneously, DNA statistical tools tend to provide more measures to the user. From a computer simulation perspective, entities in DNA are like atoms in quantum theory: they can be, though need not be, treated as probabilistic. Whereas entities in a traditional SNA model are static, entities in a DNA model have the ability to learn. Properties change over time; entities can adapt. For example, a company's employees can learn new skills and increase their value to the network, or one terrorist's death forces three more to improvise. Change propagates from one entity to the next and so on. DNA adds the critical element of a network's evolution to textual analysis and considers the circumstances under which change is likely to occur.



## Glossary

**Adjacency Network** : A Network that is a square actor-by-actor ( $i=j$ ) network where the presence of pairwise links are recorded as elements. The main diagonal, or self-tie of an adjacency network is often ignored in network analysis.

**Aggregation** : Combining statistics from different nodes to higher nodes.

**Algorithm** : A finite list of well-defined instructions for accomplishing some task that, given an initial state, will terminate in a defined end-state.

**Attribute** : Indicates the presence, absence, or strength of a particular connection between nodes in a Network.

**Betweenness** : Degree an individual lies between other individuals in the network; the extent to which an node is directly connected only to those other nodes that are not directly connected to each other; an intermediary; liaisons; bridges. It is the number of nodes a given node is indirectly connected to via its direct links.

**Betweenness Centrality** : High in betweenness but not degree centrality. This node connects disconnected groups, like a Go-between.

**Bigrams** : Bigrams are groups of two written letters, two syllables, or two words, and are very commonly used as the basis for simple statistical analysis of text.

**Bimodal Network** : A network most commonly arising as a mixture of two different unimodal networks.

**Binarize** : Divides your data into two sets; zero or one.

**Bipartite Graph** : Also called a bigraph. It's a set of nodes decomposed into two disjoint sets such that no two nodes within the same set are adjacent.

**BOM** : A byte order mark (BOM) consists of the character code U+FEFF at the beginning of a data stream, where it can be used as a signature defining the byte order and encoding form, primarily of unmarked plaintext files. Under some higher level protocols, use of a BOM may be mandatory (or prohibited) in the Unicode data stream defined in that protocol.

**Centrality** : The nearness of an node to all other nodes in a network. It displays the ability to access information through links connecting other nodes. The closeness is the inverse of the sum of the shortest distances between each node and every other node in the network.

**Centralization** : Indicates the distribution of connections in the employee communication network as the degree to which communication and/or information flow is centralized around a single agent or small group.

**Classic SNA density** : The number of links divided by the number of possible links not including self-reference. For a square network, this algorithm\* first converts the diagonal to 0, thereby ignoring self-reference (a node connecting to itself) and then calculates the density. When there are N nodes, the denominator is  $(N * (N - 1))$ . To consider the self-referential information, use general density.

**Clique** : A sub-structure that is defined as a set of nodes where every node is connected to every other node.

**Clique Count** : The number of distinct cliques to which each node belongs.

**Closeness** : Node that is closest to all other Nodes and has rapid access to all information.

**Clustering coefficient** : Used to determine whether or not a graph is a small-world network.

**Cognitive Demand** : Measures the total amount of effort expended by each agent to do its tasks.

**Collocation** : A sequence of words or terms which co-occur more often than would be expected by chance.

**Column Degree** : see Out Degree\*.

**Complexity** : Complexity reflects cohesiveness in the organization by comparing existing links to all possible links in all four networks (employee, task, knowledge and resource).

**Concor Grouping** : Concor recursively splits partitions and the user selects n splits. (n splits -> 2n groups). At each split it divides the nodes based on maximum correlation in outgoing connections. Helps find groups with similar roles in networks, even if dispersed.

**Congruence** : The match between a particular organizational design and the organization's ability to carry out a task.

**Count** : The total of any part of a Meta-Network row, column, node, link, isolate, etc.

**CSV** : "Comma Separated Value". A common file structure used in database programs for formatting output data.

**Degree** : The total number of links to other nodes in the network.

**Degree Centrality** : Node with the most connections. (e.g. In the know). Identifying the sources for intel helps in reducing information flow.

**Density** :

- **Binary Network** : The proportion of all possible links actually present in the Network.
- **Value Network** : The sum of the links divided by the number of possible links. (e.g. the ratio of the total link strength that is actually present to the total number of possible links).

**Dyad** : Two nodes and the connection between them.

**Dyadic Analysis** : Statistical analysis where the data is in the form of ordered pairs or dyads. The dyads in such an analysis may or may not be for a network.

**Dynamic Network Analysis** : Dynamic Network Analysis (DNA) is an emergent scientific field that brings together traditional Social Network Analysis\* (SNA), Link Analysis\* (LA) and multi-agent systems (MAS).

**DyNetML** : DynetML is an xml based interchange language for relational data including nodes, ties, and the attributes of nodes and ties. DyNetML is a universal data interchange format to enable exchange of rich social network data and improve compatibility of analysis and visualization tools.

**Endian** : Data types longer than a byte can be stored in computer memory with the most significant byte (MSB) first or last. The former is called big-endian, the latter little-endian. When data are exchange in the same byte order as they were in the memory of the originating system, they may appear to be in the wrong byte order on the receiving system. In that situation, a BOM would look like 0xFFFE which is a non-character, allowing the receiving system to apply byte reversal before processing the data. UTF-8 is byte oriented and therefore does not have that issue. Nevertheless, an initial BOM might be useful to identify the data stream as UTF-8.

**Entropy** : The formalization of redundancy and diversity. Thus we say that Information Entropy (H) of a text document (X) where probability p of a word x = ratio of total frequency of x to length (total number of words) of a text document.

**General density** : The number of links divided by the number of possible links including self-reference. For a square network, this algorithm\* includes self-reference (an node connecting to itself) when it calculates the density. When there are N nodes, the denominator is (N\*N). To ignore self-referential information use classic SNA\* density.

**Hidden Markov Model** : A statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and

the challenge is to determine the hidden parameters from the observable parameters.

**Homophily** : (e.g., love of the same) is the tendency of individuals to associate and bond with similar others.

- **Status homophily** means that individuals with similar social status characteristics are more likely to associate with each other than by chance.
- **Value homophily** refers to a tendency to associate with others who think in similar ways, regardless of differences in status.

**In-Degree** : The sum of the connections leading to an node from other nodes. Sometimes referred to row degree.

**Influence network** : A network of hypotheses regarding task performance, event happening and related efforts.

**Isolate** : Any node which has no connections to any other node.

**Link** : A specific relation among two nodes. Other terms also used are tie and link.

**Link Analysis** : A scientific area focused on the study of patterns emerging from dyadic observations. The relationships are typically a form of co-presence between two nodes. Also multiple dyads that may or may not form a network.

**Main Diagonal** : in a square network this is the conjunction of the rows and cells for the same node.

**Network Algebra** : The part of algebra that deals with the theory of networks.

**Meta-Network** : A statistical graph of correlating factors of personnel, knowledge, resources and tasks. These measures are based on work in social networks, operations research, organization theory, knowledge management, and task management.

**Morpheme** : A morpheme is the smallest meaningful unit in the grammar of a language.

**Multi-node** : More than one type of node (people, events, locations, etc.).

**Multi-plex** : Network where the links are from two or more relation classes.

**Multimode Network** : Where the nodes are in two or more node classes.

**Named Entity List (NEL)** : A list of ngrams that are thought to refer to specific people, organizations, or locations.

**Named-Node Recognition** : An Automap feature that allows you to retrieve proper names (e.g. names of people, organizations, places), numerals, and abbreviations from texts.

**Neighbors** : Nodes that share an immediate link to the node selected.

**NEL (project original)** : This is the named entity list auto-generated by AutoMap with AutoMap guesses as to ontology class.

***NOTE** : It may contain entities that are not true named entities and the classification may be wrong.*

***NOTE** : The size of this list is constant for a given version of automap and depends only on the tools in automap.*

**NEL (project unclassified)** : This is what remains of the NEL (project original) after named entities from the standard thesauri are removed and after named entities classified by a human are removed.

***NOTE** : The size of this list will shrink each time the NEL (project original) is processed with a new standard thesauri and new project specific classifications of named entities.*

In general, most users will do 2 to 3 passes of cleaning the NEL resulting in "additional project thesauri." If all these additions plus the standard are applied to NEL (project original) or if just the most recent addition is applied to NEL (project unclassified), the resulting NEL (project unclassified) and NEL (project classified) should be identical.

***NOTE** : Not all terms may end up being classified.*

**NEL (project classified)** : This is the set of NEL drawn from the project corpus that are classified by ontological category and have been checked for accuracy.

**NOTE :** *This includes all of n-grams in the project corpus that according to the standard thesauri are NEL.*

Checking for accuracy means either it was classified by the standard thesauri or a project user classified the term. Standard thesauri should be applied first.

**NOTE :** *The size of the NEL (project classified) should increase as more terms from the NEL (project unclassified) are classified.*

**NOTE :** *After the project is done, a CASOS person should determine which if any of the project NEL should get added to the standard thesauri.*

**Network :** Set of links among nodes. Nodes may be drawn from one or more node classes and links may be of one or more relation classes.

**Newman Grouping :** Finds unusually dense clusters, even in large networks.

**Nodes :** General things within an node class (e.g. a set of actors such as employees).

**Node Class :** The type of items we care about (knowledge, tasks, resources, agents).

**Node Level Metric :** is one that is defined for, and gives a value for, each node in a network. If there are x nodes in a network, then the metric is calculated x times, once each for each node. Examples are Degree Centrality\*, Betweenness\*, and Cognitive Demand\*.

**Node Set :** A collection of nodes that group together for some reason.

**ODBC :** (O)pen (D)ata (B)ase (C)onnectivity is an access method developed by the SQL Access group in 1992 whose goal was to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data.

**Ontology :** "The Specifics of a Concept". The group of nodes, resources, knowledge, and tasks that exist in the same domain and are connected to one another. It's a simplified way of viewing the information.

**Organization :** A collection of networks.

**Out-Degree** : The sum of the connections leading out from an node to other nodes. This is a measure of how influential the node may be. Sometimes referred to as column degree.

**Pendant** : Any node which is only connected by one link. They appear to dangle off the main group.

**Project** : The thing you are working on. This is generally associated with a research question.

**Project corpus** : The set of texts used in a specific project. These often exist in raw and in cleaned form. The cleaned form would be just .txt files.

**Random Graph** : One tries to prove the existence of graphs with certain properties by assigning random links to various nodes. The existence of a property on a random graph can be translated to the existence of the property on almost all graphs using the famous Szemerédi regularity lemma\*.

**Reciprocity** : The percentage of nodes in a graph that are bi-directional.

**Redundancy** : Number of nodes that access to the same resources, are assigned the sametask, or know the same knowledge. Redundancy occurs only when more than one agent fits the condition.

**Relation** : The way in which nodes in one class relate to nodes in another class.

**Row Degree** : see In Degree\*.

**Semantic Network** : Often used as a form of knowledge representation. It is a directed graph consisting of vertices, which represent concepts, and links, which represent semantic relations between concepts.

**Social Network Analysis** : The term Social Network Analysis (or SNA) is used to refer to the analysis of any network such that all the nodes are of one type (e.g., all people, or all roles, or all organizations), or at most two types (e.g., people and the groups they belong to).

**Specific Entity** : The name by which the person, organization or location is commonly referred to that identifies them as distinct from a generic entity. For example, John Doe is specific man is generic.

**Stemming** : Stemming detects inflections and derivations of concepts in order to convert each concept into the related morpheme.

**tfidf** : Term Frequency/Inverse Document Frequency helps determine a word's importance in the corpus. **tf (Term Frequency)** is the importance of a term within a document. **idf (Inverse Document Frequency)** is the importance of a term within the corpus.

$$tf = \frac{\text{cumulative occurrence of term } x \text{ in document } y}{\text{total number of terms in document } y}$$
$$idf = \frac{\log(\text{total number of documents in corpus})}{\text{total number of documents containing term } x}$$

$$\mathbf{tfidf = tf * idf}$$

Useful when creating a General Thesaurus.

**Thesaurus** : A list which associates multiple abstract concepts with more common concepts.

- **Generalization Thesaurus** : Typically a two-columned collection that associates text-level concepts with higher-level concepts. The text-level concepts represent the content of a data set, and the higher-level concepts represent the text-level concepts in a generalized way.
- **Meta-Network Thesaurus** : Associates text-level concepts with meta-network categories.

**Sub-Matrix Selection** : The Sub-Matrix Selection denotes which Meta-Network Categories should be retranslated into concepts used as input for the meta-network thesaurus.

**Topology** : The study of the arrangement or mapping of the elements (links, nodes, etc.) of a network, especially the physical (real) and logical (virtual) interconnections between nodes.

**Unimodal networks** : These are also called square networks because their adjacency network\* is square; the diagonal is zero diagonal because there are no self-loops\*.

**Windowing** : A method that codes the text as a map by placing relationships between pairs of Concepts that occur within a window. The size of the window can be set by the user.



## DOS Commands

### Description

A short description of some DOS commands that can be useful when using the Script.

### CD: Change Directory

#### **cd\**

Goes to the highest level, the root of the drive.

#### **cd..**

Goes back one directory. For example, if you are within the C:\Windows\COMMAND> directory, this would take you to C:\Windows>

The CD command also allows you to go back more than one directory when using the dots. For example, typing: cd... with three dots after the cd would take you back two directories.

#### **cd windows**

If present, would take you into the Windows directory. Windows can be substituted with any other name.

#### **cd\windows**

If present, would first move back to the root of the drive and then go into the Windows directory.

## **cd windows\system32**

If present, would move into the system32 directory located in the Windows directory. If at any time you need to see what directories are available in the directory you're currently in use the dir command.

## **cd**

Typing cd alone will print the working directory. For example, if you're in c:\windows> and you type the cd it will print c:\windows. For those users who are familiar with Unix / Linux this could be thought of as doing the pwd (print working directory) command.

## **DIR: Directory**

Lists all files and directories in the directory that you are currently in.

## **dir /ad**

List only the directories in the current directory. If you need to move into one of the directories listed use the cd command.

## **dir /s**

Lists the files in the directory that you are in and all sub directories after that directory, if you are at root "C:\>" and type this command this will list to you every file and directory on the C: drive of the computer.

## **dir /p**

If the directory has a lot of files and you cannot read all the files as they scroll by, you can use this command and it will display all files one page at a time.

## **dir /w**

If you don't need the info on the date / time and other information on the files, you can use this command to list just the files and directories going horizontally, taking as little as space needed.

## **dir /s /w /p**

This would list all the files and directories in the current directory and the sub directories after that, in wide format and one page at a time.

## **dir /on**

List the files in alphabetical order by the names of the files.

## **dir /o-n**

List the files in reverse alphabetical order by the names of the files.

## **dir \ /s |find "i" |more**

A nice command to list all directories on the hard drive, one screen page at a time, and see the number of files in each directory and the amount of space each occupies.

## **dir > myfile.txt**

Takes the output of dir and re-routes it to the file myfile.txt instead of outputting it to the screen.

## **MD: Make Directory**

### **md test**

The above example creates the **test** directory in the directory you are currently in.

### **md c:\test**

Create the **test** directory in the c:\ directory.

## **RMDIR: Remove Directory**

## **rmdir c:\test**

Remove the test directory, if empty. If you want to delete directories that are full, use the deltree command or if you're using Windows 2000 or later use the below example.

## **rmdir c:\test /s**

Windows 2000, Windows XP and later versions of Windows can use this option with a prompt to permanently delete the test directory and all subdirectories and files. Adding the /q switch would suppress the prompt.

## **COPY: Copy file**

### **copy \*.\* a:**

Copy all files in the current directory to the floppy disk drive.

### **copy autoexec.bat c:\windows**

Copy the autoexec.bat, usually found at root, and copy it into the windows directory; the autoexec.bat can be substituted for any file(s).

### **copy win.ini c:\windows /y**

Copy the win.ini file in the current directory to the windows directory. Because this file already exists in the windows directory it normally would prompt if you wish to overwrite the file. However, with the /y switch you will not receive any prompt.

### **copy myfile1.txt+myfile2.txt**

Copy the contents in myfile2.txt and combines it with the contents in myfile1.txt.

### **copy con test.txt**

Finally, a user can create a file using the copy con command as shown above, which creates the test.txt file. Once the above command has been typed in, a user could type in whatever he or she wishes. When you have

completed creating the file, you can save and exit the file by pressing CTRL+Z, which would create ^Z, and then press enter. An easier way to view and edit files in MS-DOS would be to use the edit command.

## **RENAME: Rename a file**

**rename c:\chope hope**

Rename the directory chope to hope.

**rename \*.txt \*.bak**

Rename all text files to files with .bak extension.

**rename \* 1\_\***

Rename all files to begin with 1\_. The asterisk (\*) in this example is an example of a wild character; because nothing was placed before or after the first asterisk, this means all files in the current directory will be renamed with a 1\_ in front of the file. For example, if there was a file named hope.txt it would be renamed to 1\_pe.txt.



Borgatti, S. P., M. G. Everett, and L. C. Freeman. (2002). UCINET for Windows, Software for Social Network Analysis: Analytic Technologies, Incorporated.

Burkart, Margaret. (1997). Thesaurus. In Marianne Buder, Werner Rehfeld, Thomas Seeger & Dietmar Strauch (Eds.), *Grundlagen der praktischen Information und Dokumentation: Ein Handbuch zur Einführung in die fachliche Informationsarbeit* (4th ed., pp. 160 - 179). München: Saur.

Carley, Kathleen M. (1993). Coding Choices for Textual Analysis: A Comparison of Content Analysis and Map Analysis. *Sociological Methodology*, 23, 75-126.

- Carley, Kathleen M. (1993). Content Analysis. In R.E. Asher & J.M.Y. Simpson (Eds.), *The Encyclopedia of Language and Linguistics* (Vol. 2, pp. 725-730). Edinburgh, UK: Pergamon Press.
- Carley, Kathleen M. (1994). Extracting Culture through Textual Analysis. *Poetics*, 22, 291-312.
- Carley, Kathleen M. (1997). Extracting Team Mental Models Through Textual Analysis. *Journal of Organizational Behavior*, 18, 533-538.
- Carley, Kathleen M. (1997). Network Text Analysis: The Network Position of Concepts. In Carl W. Roberts (Ed.), *Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts* (pp. 79-100). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Carley, Kathleen M. (2002). Smart Agents and Organizations of the Future. In Leah Lievrouw, and Sonia Livingstone (Ed.), *The Handbook of New Media* (pp. 206-220). Thousand Oaks, CA: Sage.
- Carley, Kathleen M. (2003). Dynamic Network Analysis. In Ronald Breiger, Kathleen Carley & Philippa Pattison (Eds.), *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, Committee on Human Factors* (pp. 133-145). Washington, DC: National Research Council.
- Carley, Kathleen M., Diesner, Jana, Reminga, Jeffrey, & Tsvetovat, Maksim. (2007). Toward an Interoperable Dynamic Network Analysis Toolkit. *Decision Support Systems: Special Issue Cyberinfrastructure for Homeland Security*, 43(4), 1324-1347.
- Carley, Kathleen M., and David Kaufer. (1993). Semantic Connectivity: An Approach for Analyzing Semantic Networks. *Communication Theory*, 3(3), 183-213.
- Carley, Kathleen M., and Michael Palmquist. (1992). Extracting, Representing and Analyzing Mental Models. *Social Forces*, 70(3), 601-636.
- Carley, Kathleen M., & Reminga, Jeffrey. (2004). ORA: Organizational Risk Analyzer. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Institute for Software Research.
- Diesner, Jana, & Carley, Kathleen M. (2004). AutoMap 1.2 : Extract, Analyze, Represent, and Compare Mental Models from Texts.

Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Institute for Software Research.

- Diesner, Jana, & Carley, Kathleen M. (2005, April 21-23). *Exploration of Communication Networks from the Enron Email Corpus*. Paper presented at the SIAM International Conference on Data Mining: Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA.
- Diesner, Jana, & Carley, Kathleen M. (2005). Revealing Social Structure from Texts: Meta-Matrix Text Analysis as a novel method for Network Text Analysis. In V.K. Narayanan & D.J. Armstrong (Eds.), *Causal Mapping for Information Systems and Technology Research: Approaches, Advances, and Illustrations* (pp. 81-108). Harrisburg, PA: Idea Group Publishing.
- Diesner, Jana, Carley, Kathleen M., & Katzmaier, Harald. (2007, May 1-6). *The morphology of a breakdown. How the semantics and mechanics of communication networks from an organization in crises relate*. Paper presented at the XXVII Sunbelt Social Network Conference, Corfu, Greece.
- Diesner, Jana, Kumaraguru, Ponnurangam, & Carley, Kathleen M. (2005). *Mental Models of Data Privacy and Security Extracted from Interviews with Indians*. Paper presented at the 55th Annual Conference of the International Communication Association (ICA), New York, NY.
- Diesner, Jana, & Stuetzer, Cathleen. (2008, July 24). *Relationen finden/Finding Relations*. Paper presented at the Kunstsammlungen Chemnitz, Chemnitz Art Collections.
- Jurafsky, Daniel, & Marton, James H. (2000). *Speech and Language Processing*. Upper Saddle River, New Jersey: Prentice Hall.
- Kaufers, David, and Kathleen M. Carley. (1993). Condensation Symbols: Their Variety and Rhetorical Function in Political Discourse. *Philosophy and Rhetoric*, 26(3), 201-226.
- Klein, Harald. (1996). Classification of Text Analysis Software. In Rudiger Klar & Otto Opitz (Eds.), *20th Annual Conference of the Gesellschaft für Klassifikation e.V.* (pp. 255-261). University of Freiburg: Springer. Krovetz, Robert. *Word Sense Disambiguation for Large Text Databases*. Unpublished PhD Thesis, University of Massachusetts, 1995.

- Magnini, Bernardo, Negri, Matteo, Prevete, Roberto, & Tanev, Hristo. (2002). A Wordnet-based Approach to Named-Entities Recognition. *SemaNet'02: Building and Using Semantic Networks* (pp. 38-44). Taipei, Taiwan.
- Mrvar, Andrej. (2004). Measures of Centrality and Prestige, from <http://mrvar.fdv.uni-lj.si/sola/info4/uvod/part4.pdf>
- Palmquist, Michael, Kathleen M. Carley, and Thomas Dale. (1997). Two applications of automated text analysis: Analyzing literary and non-literary texts. In C. Roberts (Ed.), *Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts* (pp. 171-189). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Popping, R., & Roberts, C.W. (1997). Network Approaches in Text Analysis. In R. Klar & O. Opitz (Eds.), *20th Annual Conference of the Gesellschaft für Klassifikation e.V.* (pp. 381-898). University of Freiburg: Springer.
- Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.
- Shannon, Claude E., & Weaver, Warren. (1949). *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press.
- Tsvetovat, Maksim, Reminga, Jeffrey, & Carley, Kathleen M. (2004). DyNetML: Interchange Format for Rich Social Network Data. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Institute for Software Research. From <http://reports-archive.adm.cs.cmu.edu/anon/isri2004/abstracts/04-105.html>
- Wasserman, Stanley, & Faust, Katherine. (1994). *Social Network Analysis: Methods and Applications*. Cambridge: University of Cambridge Press.
- Zuell, Cornelia, & Alexa, Melina. (2001). Automatisches Codieren von Textdaten. Ein Ueberblick ueber neue Entwicklungen. In Werner Wirth & Edmund Lauf (Eds.), *Inhaltsanalyse - Perspektiven, Probleme, Potenziale* (pp. 303-317). Koeln: Herbert von Halem.



## Data Collection

### Description

AutoMap is designed to extract, analyze, and interpret relational data (also known as network data) from unstructured, natural language text data.

### Relation Extraction Sources

The source of your data can be anything: books, television, newspapers, blogs, emails, internet sites. AutoMap will extract the data and sort it into relational data which can be further analyzed in ORA.

### Method

The first thing to do is identify the problem/goal.

Next all/some of the concepts need identified in the texts and the links between them (binary, typed, directed, weighted) can be defined.

Now this data can be represented as relational data (graph or list).

Then the data can be analyzed.

And finally the results can be interpreted.

### How is network data collected?

Interviews, Automated (web-based surveys).

Person	Albert	Betty	Charlie
Albert	0	1	0
Betty	0	0	1
Charlie	0	1	0

Data collection is more of an approximation via Network Text Analysis as most real-world networks and sequential data are not iid (independent and

identically distributed). Network data is a concise representation of what's in the text data - Is it not the truth, only an approximation.



AutoMap is a natural language processing system. It is used as a means to understand text, or to process text to be used in conjunction with other tools such as the CASOS \*ORA program. Some of the ways in which AutoMap is used:

1. To extract a metanetwork representation of a dynamic/social network as expressed in text.
2. To extract a semantic network to understand the relationships between concepts in texts.
3. To clean and process text files for example by removing symbols and numbers, deleting unnecessary words, and stemming.
4. To identify concepts and the frequency of concepts appearing in texts.

## Description

The AutoMap GUI (Graphical User Interface) contains access to AutoMap's features via the menu items and shortcut buttons. The purpose of the GUI is to aid in the exploration of processing steps. Users will be able to understand the impact of processing parameters and processing order.

The processing of an extensive collection of texts is best done using the script version of AutoMap. The same processing steps available in the AutoMap GUI are available in the AutoMap Script.

## Guide Roadmap

- A. Interface Overview
- B. Tutorial 1: Creating Concept and Union Concept List
- C. Tutorial 2: Using Delete Lists

D. Tutorial 3: Content Analysis to Semantic Network

E. Interface Details

## The User Interface Overview

### The Pull Down Menus

The **Text Display Window** displays the text file as it appears based on the preprocessing that has been applied to it. The **File Navigation Buttons** allow you to move between individual text files. The **Filename Box** will identify the name of the currently displayed text file.

The **Message Window** will provide feedback. The **Quick Launch Buttons** are the most commonly used menu commands, placed in the main window for quick access.

The **File Menu** contains loading and saving commands, and exit, to quit the AutoMap program.

The **Edit Menu** contains configuration options.

The **Preprocess Menu** contains commands that will modify the text file. These commands may be applied in any order. The result of the preprocessing is displayed in the Text Display Window, with the name of the preprocessing step displayed in the Preprocess /Order Window.

The **Generate Menu** contains commands for generating end results. The output of these commands may be created to be used as input to other programs. For instance, a generated MetaNetwork DyNetML file can be used as input to \*ORA for analysis.

The **Tools Menu** contains launchable external tools. These tools are provided to aid in the editing of supplemental files or the viewing of end results. AutoMap uses standard file formats such as text (.txt), comma separated value (.csv) or XML (.xml) in order to provide maximum interaction with other tools.

The Help Menu contains the AutoMap help system.

## Before You Begin

AutoMap is a system that starts with text files. Before being able to use the features of AutoMap, it is necessary to have text to process. This text can be obtained from email, news articles, publications, web pages, or text typed in using a text editor.

AutoMap will process all text (.txt) files in a directory. It is not necessary to combine text into a single file. Some larger text files can be split into smaller text files to do analysis of sections individually.

You will be prompted for the location of where to store the files that are the results of your processing. Many people will create a folder to keep the text files and all of the results. In this work folder, create a subfolder to store the original texts and additional subfolders to store the results you will generate.

For example, if we are interested in only creating concept lists from our texts, we can create the following file structure:

```
C:\Mike\working
```

```
C:\Mike\working\texts
```

```
C:\Mike\working\concepts
```

When generating a concept list, be sure to navigate to the appropriate folder, such as C:\Mike\working\concepts folder in our example, to store the results.

## Simple Tutorials

### Creating Concept & Union Concept Lists

#### Description

**Concept Lists & Union Concept Lists** compile lists based on individual and multiple files giving their frequency. A Concept List collects concepts in one file only. Union Concept Lists collect concepts from all currently loaded files.

## Step 1: Load Text Files

From the Pull Down Menu select *File => Select Input Directory*. Navigate to a directory with your text files and click Select.

## Step 2: Create a Concept List

From the Pull Down Menu select *Generate => Concept List*. Navigate to a directory to save the list and click **Select**. If you have other files in that directory, you will be alerted that some files may be overwritten. As long as you did not add or remove input files from a previous run there is no problem as the previous concept list files will be overwritten with the new concept list files. The file name will be the same as the original text file, substituting the.txt for.csv. For instance mike.txt as an input text file will create a concept list file named mike.csv.

AutoMap will ask if you want to generate a **Union Concept List**. It is a good idea to create this list. All files in the directory you select to save your concept lists in will be used to create the union concept list. If you have old concept lists in there not from the current run, they will also be used.

## Viewing a Concept List

From the Pull Down Menu select *Tools => Concept List Viewer*. From the Viewer Pull Down Menu select *File => Open File*. Navigate to the directory where your Concept Lists are stored and select one and click **Open**. If a **Concept List** is chosen only the concepts from one file are displayed. If a **Union Concept List** is chosen it will display concepts from all files. As the concept lists are saved in a standard.csv format, you can also view them in a text editor or a spreadsheet program such as Microsoft Excel.

## Creating a Delete List

From the viewer menu you can create a Delete List by placing a check mark in the **Selected** columns then from the Pull Down Menu select *File => Save as Delete List*. Navigate to the directory, type in a new file name, and click **Open** to save your new Delete List.

## Comparing Files

You can also compare the currently loaded file with another using **File => Compare File**. Navigate to the file to compare the first file with and click **Open**.

AutoMap will color code the concepts: no color means the information is the same in both the original and compared files, **red** means the concept was in the original but not in the compared file, **green** means the concept was not in the original but is in the compared file, and **yellow** the concepts are the same but the data (such as frequency) has changed.

## Using Delete Lists

### Description

**Delete Lists** allow you to remove **non-content** bearing conjunctions, articles and other noise from texts. Delete List can be created internally in AutoMap or externally in a text editor. The list itself is a text file that contains a list (one concept per line) of the words to be deleted from the text.

**NOTE :** *Whether you apply the Delete List(s) before or after applying a Thesauri will depend on your exact circumstances.*

## Step 1. Create a Delete List

There are two ways to create a new delete list:

### *Within AutoMap*

Use the Concept List Viewer by select **Tools => Concept List Viewer**. Place a check mark next to the concepts to include. Form the view menu select **File => Save as Delete List**. The Delete List created can be viewed in the Delete List Editor by selecting **Tools => Delete List Editor**.

### *Outside of AutoMap*

Using a text editor or spreadsheet program capable of saving output as.txt files to manually create a Delete List. The main rule is one concept per line.

**NOTE :** *Delete Lists can be opened in Excel, worked with, and then re-saved as a.txt file.*

## Step 2. Load Text Files

From the Pull Down Menu select **File => Select Input Directory**. Navigate to a directory with your text files and click **Select**.

## Step 3. Apply a Delete List

From the Pull Down menu select **Preprocess => Apply Delete List**. Navigate to the file that contains your delete list and click **Select**.

## Step 4. Select Type of Deletion

You will be prompted for the type of delete to perform. Direct will remove the concept entirely, whereas Rhetorical will replace the concept with xxx. Make your selection and click **OK**.

## The Results

The results will appear in the Text Display Window.

## Using a Generalization Thesaurus

### Description

To use a unified key concept to represent many varieties of the same concept. For example to replace a contraction "don't" with its individual words "do not". This would be represented in the file as:

`don't, do not`

Be sure there are no extra spaces around the comma as they will be used in the translation. A spreadsheet program will not put in extra spaces.

## Step 1. Review Your texts

Read through your texts to identify concepts to place into your thesaurus.

## Step 2. Create a Thesaurus

You can create a thesaurus in either a text editor or a spreadsheet program that can save files as.csv files. The format of an entry is **concept,key\_concept**. Concept can be single or multiple words and key\_concept is one set of words usually separated by underscores.

US,United\_States  
United States,United\_States

## Step 3. Load Text Files

Place all your files in the same directory. Make sure that directory is empty before placing the files. From the Pull Down Menu select **File => Select Input Directory**. Navigate to a directory with your thesaurus file and click **Select**.

## Step 4. Apply Thesaurus

From the Pull Down Menu select **Preprocess => Apply Generalization Thesauri**. Navigate to a directory with your thesauri and click **Select**. The results will be displayed in the Text Display Window.

## Content Analysis to Semantic Network

### Description

A semantic network will identify the relationships between concepts in the text.

## Step 1. Load Text Files

Place all your files in the same directory. Make sure that directory is empty before placing the files. From the Pull Down Menu select **File => Select Input Directory**. Navigate to a directory with your text files and click **Select**.

## (Optional) Step 2. Create Concept Files

From the Pull Down Menu select **Generate => Concept List**. Navigate to the directory to store these files (should be an empty directory) and click **Select**. AutoMap will ask if you want to create a Union Concept List. This will be useful for creating a Delete List on multiple files therefore click **Yes**.

## (Optional) Step 3. Build a Generalization Thesauri

Review your texts for single concepts under multiple instances. (e.g., U.S. and United States can both be turned into United\_States). In a text editor create an csv file with a list of entries consisting of a concept (one or more words in a file) and the new concept (all one string of words usually connected with an underscore) separated by a comma (e.g. U.S.,United\_States and United States,United\_States).

After constructing this file save it to a directory.

## (Optional) Step 4. Apply a Generalization Thesauri

From the Pull Down Menu select **Preprocess => Apply Generalization Thesauri**. Navigate to the directory containing your new thesaurus file, select a thesaurus, and click **Select**.

## (Optional) Step 5. Build a Delete List

Open the Union Concept List with **Tools => Concept List Viewer**. Place a check mark next to each concept you want placed in the Delete List. From the Pull Down Menu select **File => Save Delete List** and navigate to where you want to save it.

## (Optional) Step 6. Apply a Delete List

From the Pull down Menu select *Preprocess => Apply Delete List*. Navigate to the directory containing your delete List, highlight the file, and click **Select**. The preprocessed files will display in the Text Display Window.

### *Adjacency*

When applying a delete list AutoMap will inquire as to the type of adjacency to use. The **Adjacency option** determines whether AutoMap will replace deleted concepts with a placeholder or not.

- **Direct Adjacency** : Removes concepts in the text that match concepts specified in the delete list and causes the remaining concepts to become adjacent.
- **Rhetorical Adjacency** : Removes concepts in the text that match concepts specified in the delete list and replaces them with **(xxx)**. The placeholders retain the original distances of the deleted concepts. This is helpful for visual analysis.

The newly pre-processed texts can be viewed in the main window.

## Step 7. Create a Semantic Network

From the Pull Down Menu select *Generate => Semantic Network*. AutoMap will generate one XML file for each text loaded for use in ORA. Navigate to the directory to save these files and click Select.

AutoMap will output one XML file for each text file loaded. AutoMap will ask a couple of questions as to how you want to format the DyNetML file. You will be asked to select **Directionality** (Unidirectional or Bidirectional), **Window Size** (maximum distance between two concepts to be connected), **Stop Unit** (Clause, word, sentence, or paragraph), and **Number of [Stop Units]**.

## Step 8. Load the DyNetML files in \*ORA

Start \*ORA and load the newly created **XML** files \*ORA.

## Multiple Delete Lists and Thesauri

Multiple delete lists and thesauri can be applied to the same text by loading, and applying the first delete list then loading, and applying a subsequent delete list. Any number can be applied in this manner. They can be viewed in order using the Pull Down Menu in the menu bar.

## Un-apply a Delete List or a Thesaurus

Delete Lists and Thesauri can be **unapplied** but only in the same order that all preprocessing has been applied. If other preprocessing steps have been taken then you must Undo those steps also.

## Modifying a Delete List

After a Delete list is created you can modify it using the **Delete List Editor**. From the Pull Down menu select *Tools => Delete List Editor*. From the Viewer's Pull Down Menu select *File => Open File* and navigate to the directory containing your Delete Lists. Place a check mark in the **Select to Remove** column for concepts to remove from the Delete List. Typing concepts into the textbox and clicking **[Add Word]** will add concepts to the Delete List. When you are finished select *File => Save as Delete List*.

## Save text(s) after Delete List

You can save your texts after applying a delete list by selecting from the Pull Down Menu *File => Save Preprocess Files*. This must be done before any other further preprocessing is performed as this option saves the texts at the highest level of preprocessing.

## Interface Details

### The Pull Down Menu

#### File

*File => Select Input Directory* loads all text files into AutoMap from the directory chosen. All.txt files in the directory will be loaded.

*File => Import Text* is similar to Select Input Directory as it loads all.txt files from one directory but provides additional support to load text files in other encodings. The default is **Let AutoMap Detect**.

*File => Save Preprocessed Text Files* saves all your files based on the highest level of preprocessing.

*File => Exit* will exit the AutoMap GUI program.

## Edit

*Edit => Set Font* allows the user to change the font of the **Display Window**. The importance of changing the font is to display foreign character text. The font choices are based on the fonts available on the computer.

## Preprocess

These options permit the cleaning and modification of the text in preparation of generating output. Contains the following preprocessing options: **Remove Extra Spaces, Remove Punctuation, Remove Symbols, Remove Numbers, Convert to Lowercase, Convert to Uppercase, Apply Stemming, Apply Delete List, & Apply Generalization Thesauri**.

These functions alter the text. They may be applied in any order as there should be no side effects.

## Generate

Used for the generation of output from preprocessed files. The following output are available: **Concept List, Semantic List, Parts of Speech Tagging, Semantic Network, DyNetML MetaNetwork, Bigrams, Text Properties, Named entities, Feature Selection, Suggested MetaNetwork Thesauri, Union Concept Lists**.

These functions output files and are based on the highest level of preprocessing done.

## Tools

AutoMap contains a number of Editors and Viewers for the files. These include: **Delete List Editor, Thesauri Editor, Concept List Viewer, Semantic List Viewer, DyNetML Network Viewer.**

These allow the user to edit support files used in preprocessing, or to view the results that have been generated.

## Help

The Help file and about AutoMap.

## Quick Launch Buttons

These buttons correspond to the functions in the Preprocess Menu.

## File Navigation Buttons

Used to display the files in the main window. The buttons contain from left to right: **First, Previous, Goto, Next, and Last.**

## Preprocess Order Window

Contains a running list of the preprocesses performed on the files. This can be undone one process at a time with the Undo command. The Undo affects the latest preprocess only.

## Filename Box

Displays the name of the currently active file. Using the File Navigation Buttons will change this and as well as the text displayed in the window.

## Text Display Window

Display the text for the file currently listed in the Filename Box.

# Message Window

Area where AutoMap display the actions taken as well errors encountered.



The AM3Script is a command line utility that processes large numbers of files using a set of processing instructions provided in the configuration file. Some of the ways in which AutoMap is used:

- To extract a metanetwork representation of a dynamic/social network as expressed in text.
- To extract a semantic network to understand the relationships between concepts in texts.
- To clean and process text files for example by removing symbols and numbers, deleting unnecessary words, and stemming.
- To identify concepts and the frequency of concepts appearing in texts.

## Description

AM3Script uses tags to tell AutoMap which functions to access. Functions are performed in the order they are listed in the config file. All preprocessing functions are followed by all processing functions and finally all post-processing functions are performed. Necessary output files are also written depending on the tags used in the config file.

If working with large numbers of texts it is best to use the script version as opposed to the GUI. The same processing steps available in the AutoMap GUI are available in the AutoMap Script.

## Guide Roadmap

A. Script Overview

B. Tag List

C. Tutorial 1: Setting up a run in the Script

D. Tutorial 2: Using Delete Lists

E. Tutorial 3: Using a Thesauri

## Before You Begin

AutoMap is a system that starts with text files. Before being able to use the features of AutoMap, it is necessary to have text to process. This text can be obtained from email, news articles, publications, web pages, or text typed in using a text editor.

AM3Script will process all text (.txt) files in a directory. It is not necessary to combine text into a single file. Some larger text files can be split into smaller text files to do analysis of sections individually.

It is suggested the user create sub-directories for input files, output, and support files all within an project directory. This assists in finding the correct files later and prevents AutoMap from overwriting previous files.

```
C:\My Documents\dave\project\input  
C:\My Documents\dave\project\output  
C:\My Documents\dave\project\support
```

Be sure to create the correct pathway in your config files to assure your files are written into the correct directory.

## Running AutoMap Script

Once the configuration file has been created, the AM3Script is ready to use. The following is a brief on running the script.

1. Create a new .aos file. Configure the AM3Script .aos file as necessary by selecting the tags to use (Tag explanations in next section). Be sure to include pathways to input and output directories. Be sure to name the config file something unique.

```
<Settings>  
<AutoMap  
textDirectory="C:\My Documents\dave\project\input"  
tempWorkspace="C:\My Documents\dave\project\output"  
textEncoding="unicode"/>  
</Settings>
```

2. Open a Command Prompt Window
3. Navigate to where the AutoMap3 program is installed. Mine is in Program Files. Yours could be in a different location.

e.g. `cd C:\Program Files\AM3`

4. To run AM3Script type the following at the command prompt:

```
am3script project.aos
```

**NOTE :** *project.aos is the name of my config file. Substitute the name of your config file. Also make sure there is a space between am3script and the name of your file.*

5. AM3Script will execute using the .aos file specified.

## For Advanced Users

It is possible to set the your PATH environmental variable to include the location of the install directory so that AM3Script can be used in any directory from the command line. Please note this is not recommended for users that have no experience modifying the PATH environmental variable.

## Script name

The script.aos file can be named whatever you like but we do recommend keeping the **.aos** suffix. This way you can do multiple runs to the files in a concise order: **step1.aos, step2.aos, step3.aos.**

## Pathways (relative and absolute)

AM3Script config files allow you to specify pathways as either relative or absolute. It's important to know the difference. For relative pathways AutoMap always starts at the location of the AM3Script file. You can go up a directory with (`..\`) or down into a directory (`\aDirectory`). The last parameter will be the filename to use.

AM3Script resides in the directory where AutoMap was installed.

The pathway `..\input\aTextFile.txt` tells AutoMap to **go up one directory** then down into the **input** directory and find the file **aTextFile.txt**.

The pathway **C:\My Documents\dave\input\ATextFile.txt** tells AutoMap to start at the root directory of the hard drive and follow the designated pathway to the file.

**NOTE :** *If given a non-existent pathway you will receive an error message during the run.*

## Tag Syntax in AM3Script

There are two styles of tags in the AM3Script. The first one uses a set of two tags. The first tag starts a section and the second tag ends the section. The second tag will contain the exact same word as the first but will have, in addition, a "/" appended after the word and before the ending bracket. This designates it as an ending tag. All the parameters/attributes pertaining to this tag will be set-up between these two tags. e.g. **<aTag></aTag>**.

The second style is the self-ending tag as it contains a "/" within the tag. Any attributes used with this tag are contained within the tag e.g. **<aTag attribute="attributeName"/>**.

## Output Directory syntax (TempWorkspace)

Output directories created within functions under the <PreProcessing> tag will all be suffixed with a number designating the order they were performed in. If a function is performed twice, each will have a separate suffix e.g. Generalization\_3 and Generalization\_5 denotes a Generalization Thesauri was applied to the text in the 3rd and 5th steps. Using thesauriLocation different thesauri could be used in each instance. For all other functions outside PreProcessing there is no suffix attached.

**NOTE :** *The output directories specified above are in a temporary workspace and the content will be deleted if AM3Script uses this directory again in processing. It is recommended that the directory specified in the temp workspace be an empty directory. Also, for output that user wishes to keep from processing it is recommended to use the outputDirectory parameter within the individual processing step.*

## Example

```
<AddAttributes3Col attributeFile="C:\My
Documents\dave\project\attributeFiles\attributes.txt"
outputDirectory="C:\My
Documents\dave\project\3ColAttribute\" />
```

By using these tags the user can specify where they want the individual processing step output to go. It also makes finding the location of the output files much simpler instead of looking through the contents of the TempWorkspace.

## AM3Script Tags Details

### <Script> </Script> (required)

This set of tags is used to enclose the entire script. Everything used by the script must fall between these two tags. The only line found outside these tags will be the declaration line for xml version and text-encoding information: `<?xml version="1.0" encoding="UTF-8"?>`

### <Settings> </Settings> (required)

Used for the setting for the default directories for text and workspace. For AM3Script the tag is `<AutoMap/>`

**NOTE :** *Any of the parameters can use `inputDirectory` and `outputDirectory` to override the default file location. These pathways will be relative to the location of the AM3Script.*

### <AutoMap/> (Required)

The `<AutoMap/>` tag contains default pathways used by all functions and the type of text encoding to use. Any function can override these pathways by setting `inputDirectory` and `outputDirectory` within it's own tag. The location of text files to process is contained in `textDirectory="C:\My Documents\dave\project\input"`. The location of the files that will be written to the output directory is in **`class="sometext">tempWorkspace="C:\My Documents\dave\project\output"`**. To specify the encoding method to use set **`textEncoding="unicode"`** (currently UTF-8 is the default. AutoMap uses UTF-8 for processing. Please make sure to set

text encoding to your correct specification of your text.). **AutoDetect** will attempt to detect and convert your text over to UTF-8.

## <Utilities> </Utilities> (required)

The <Utilities> tag contains the sections <PreProcessing>, <Processing>, and <PostProcessing>. All three sections need to be nested within the <Utilities> tag in that order.

## AutoMap 3 Preprocessing Tags

### <PreProcessing> </PreProcessing> (required)

These are utilities that modify raw text. The order the steps are placed in the file is the order they are performed. You can also perform any of these utilities multiple times. e.g. perform a <Generalization/>, then a <DeleteList/>, then another <Generalization/>. Each step's results will be written to a separate output directory.

### <RemoveNumbers/>

This parameter accepts either **whiteOut="y"** or **whiteOut="n"**. A "y" replaces numbers with spaces i.e. C3PO => C PO. A "no" removes the numbers entirely and closes up the remaining text e.g. C3PO => CPO.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<RemoveNumbers whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
```

```
</Utilities>
</Script>
```

## <RemoveSymbols/>

This parameter accepts either **whiteOut="y"** or **whiteOut="n"**. A "y" replaces symbols with spaces. A "no" removes the symbols entirely and closes up the remaining text. The list of symbols that are removed: ~`@#\$\$%^&\* \_+={}[]\|/;<>.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<RemoveSymbols whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <RemovePunctuation/>

This parameter accepts either **whiteOut="y"** or **whiteOut="n"**. A "y" replaces punctuation with spaces. A "no" removes the punctuation entirely and closes up the remaining text. The list of punctuation removed is: ., : ; ' " ( ) ! ? - .

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
```

```
RemovePunctuation whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <RemoveExtraWhiteSpace/>

Find instances of multiple spaces and replaces them a single space. Note, there are no extra parameters for this step. It's only function is to reduce multiple spaces to one space.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
RemoveExtraWhiteSpace />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <Generalization/>

The Generalization Thesaurus are used to replace possibly confusing concepts with a more standard form. e.g. a text contains both `United States` and `U.S.` The Generalization Thesaurus could have two entries which replace both the original entries with `united_states`.

If **useThesauriContentOnly="n"** AutoMap replaces concepts in the Generalization Thesaurus but leaves all other concepts intact. If **useThesauriContentOnly="y"** then AutoMap replaces concepts but removes all concepts not found in the thesaurus.

The other parameter is **thesauriLocation**. This allows you to specify the pathway to the thesaurus file to use.

The questions now is whether to use one big thesaurus or several smaller thesauri. When trying to replicate results over many runs using one file is easier to replicate.

The order of the thesauri entries will skew the results. (e.g. if you have both John & John Smith you need to put John Smith first. If John is listed first the end result will be John\_Smith\_Smith.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
Generalization thesauriLocation="C:\My Documents\dave\project\support\thesauri.csv"
useThesauriContentOnly="y" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <DeleteList/>

The Delete List is a list of concepts (one concept per line) to remove from the text files before output file. Set **adjacency="d"**, for direct (removes the space left by deleted words) and remaining concepts now become "adjacent" to each other. Set **adjacency="r"** for rhetorical (removes the concepts but inserts a spacer within the text to maintain the original distance between concepts).

The other parameter is **deleteListLocation** which specifies the pathway to the Delete List.

```
<Script>
<Settings>
<AutoMap
```

```

textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
DeleteList adjacency="r" deleteListLocation="C:\My
Documents\dave\project\support\deleteList.txt" saveTexts="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <FormatCase/>

FormatCase changes the output text to either "lower" or "upper" case. If **changeCase="l"** then AutoMap will change all text to lowercase. **changeCase="u"** changes all text to uppercase.

```

<Script>
<Settings>
<AutoMap>
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<FormatCase changeCase="u"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <Stemming/>

Stemming removes suffixes from words. This assists in counting similar concepts in the singular and plural forms (e.g. plane and planes). These concepts would normally be considered two terms.

After stemming planes becomes plane and the two concepts are counted together.

There are two stemming options: **type="k"** uses the KSTEM or Krovetz stemmer and **type="p"** uses the Porter stemmer.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Stemming type="k" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <Processing> (required)

These steps are performed after all "Pre-Processing" is finished. They are performed in the order they appear in the AM3Script.

## <POSExtraction/>

**posType="ptb"** specifies a tag for each part of speech.

**posType="aggregate"** groups many categories together using fewer Parts-of-Speech tags.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
```

```
<posType="ptb" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <Anaphora />

An anaphoric expression is one represented by some kind of deictic, a process whereby words or expressions rely absolutely on context. Sometimes this context needs to be identified. These definitions need to be specified by the user. Used primarily for finding personal pronouns, determining who it refers to, and replacing the pronoun with the name.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<posType="ptb" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

**NOTE :** For Anaphora to work POS must be run first.

## <ConceptList />

Creates a separate list of concepts for each loaded text file. A Delete List or Generalization Thesauri can be performed before creating these lists to reduce the number of concepts needed to be included in this file. These concept Lists can be loaded into a spreadsheet and sorted by any of the headers.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<ConceptList />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <SemanticNetwork/>

A semantic network displays the connection between a text's concepts. These links are defined by four parameters. **windowSize**: the distance two concepts can be apart and have a relationship. textUnit defined as (S)entence, (W)ord, (C)lause, or (P)aragraph. resetNumber defines the number of textUnits to process before resetting the window. directional defined as Unidirectional (which looks forward only in the text file) or Bi-Directional (which finds relationships in either direction).

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
```

```

<Processing>
<SemanticNetwork windowSize="2 textUnit="S" resetNumber="2" directional="U" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <MetaNetwork />

This associates text-level concepts with Meta-Network categories (e.g. agent, resource, knowledge, location, event, group, task, organization, role, action, attributes, when). Concepts can be translated into multiple Meta-Network categories. `thesauriLocation="C:\My Documents\dave\project\support"` designates the location of the MetaNetwork Thesauri, when used.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
MetaNetwork thesauriLocation="C:\My Documents\dave\project\support\thesauri.csv" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <UnionConceptList />

Union Concept Lists is a list of concepts taken from all texts currently loaded, rather than only one text file. It reports total frequency, related frequency, and cumulative frequencies of concepts in all text sets. It's helpful in finding frequently occurring concepts over all loaded texts.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<UnionConceptList />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

**NOTE :** *The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.*

## <NGramExtraction />

NGramExtraction creates a file listing all the NGrams, their frequency in the files, their relative frequency to each other, and the gram type.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<NGramExtraction />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>

```

```
</PostProcessing>
</Utilities>
</Script>
```

## <SemanticNetworkList />

Creates a file consisting of pairs of concepts and their frequency within the text files. This takes four parameters: windowSize: the distance two concepts can be apart and have a relationship. textUnit defined as (S)entence, (W)ord, (C)lause, or (P)aragraph. resetNumber defines the number of textUnits to process before resetting the window. directional defined as Unidirectional (which looks forward only in the text file) or Bi-Directional (which finds relationships in either direction).

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<SemanticNetworkList directional="U" resetNumber="1" textUnit="S" windowSize="5"
/>
</PreProcessing>
</Processing>
</PostProcessing>
</Utilities>
</Script>
```

## <PostProcessing> </PostProcessing> (required)

The PostProcessing section contains functions to perform after all Processing steps are complete.

```
<addAttributes>
```

Additional attributes can be added to the nodes within the generated DyNetML file. `attributeFile="C:\My Documents\dave\project\support\attribute_file"` is the pathway to the file which contains a header row with the attribute name.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<addAttributes attributeFile="C:\My Documents\dave\project\support\attribute_file" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

This is similar to `<addAttributes>` but uses name and value.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<addAttributes3Col attributeFile="C:\My Documents\dave\project\support\3Col_file" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

**<UnionDyNetml />**

UnionDyNetml creates a union of all dyNetml in a specified directory. It requires a unionType which is s or m. "s" is for a union of semantic networks and "m" is for metanetworks.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<UnionDyNetml unionType="s" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## Simple Tutorials

### Setting Up and Using Thesauri

#### Description

Thesauri are used to reduce the number of unique concepts in the texts by assigning a key concept to multiple versions of the same concept. This example uses the file structure below. Your file structure may differ

```
C:\My Documents\dave\project\input
C:\My Documents\dave\project\output
C:\My Documents\dave\project\support
```

#### Step 1: Examining the text

If you know the subject matter then many of the multiple versions of a concept will be known already. Other times it will be necessary to examine the text to determine what those concepts are.

Ted is a U.S. citizen. He lives in the United States. Ted says, I love living in America.

There are three concepts that all mean the same thing: U.S., the United States, and America.

## Step 2: Creating a Thesauri

Once the multiple word concepts are identified you can create a thesauri to combine them into key concepts. Remember:

1. One concept per line
2. Concept and key concept separated by a comma (no spaces before or after the comma)
3. Concept can be multiple words
4. Key concept can only be one word but may contain dividing punctuation (underscores are mainly used for this purpose).

```
U.S.,the_United_States_of_America  
the United States,the_United_States_of_America  
America,the_United_States_of_America
```

Save this file as a .csv file.

## Step 3: Using in the .aos file

Place the tag `<Generalization thesauriLocation="C:\My Documents\dave\support\genThes.csv" useThesauriContentOnly="y">` in the `<PreProcessing>` section. Select whether to use thesauri content only: y (make thesaurus replacements but output only the concepts listed in the thesaurus) or n (no: make thesaurus replacements but output all concepts). Place the pathway to your newly created Thesaurus in the `thesauriLocation` parameter.

## Step 4: Run the script

Open a Command Run window and navigate to the directory where AutoMap3 was installed. At the prompt type `am3script project.aos` file. When finished navigate to the output directory denoted in the .aos file to find your output files.

## Step 5: View the Results

Open the newly created text files in a text editor to review.

## Setting Up and Using Delete Lists

### Description

Delete Lists can be created using a text editor or spreadsheet program.

### Step 1: Creating Delete Lists with a text editor

Open your text editor or spreadsheet and create a list of concepts to use as a Delete List. Place only one, single word concept per line. Save as a .txt file.

### Step 2. Make a new .aos file

Make a copy of the standard .aos file and open it in a text editor. Specify where your input are and where to write the output files

```
textDirectory="C:\My Documents\dave\project\input"  
tempWorkspace="C:\My Documents\dave\project\output"
```

Save this new .aos file in the same directory as the AM3Script file. If the file is not in the same directory then AM3Script will fail.

### Step 3: Using in the .aos file

Place the tag `<DeleteList adjacency="" deleteListLocation="">` in the `<PreProcessing>` section. For adjacency select d (direct: totally remove deleted concepts) or r (rhetorical: replace deleted concepts with a placeholder). Place the pathway to your newly created Delete List in the `deleteListLocation` parameter.

```
<DeleteList adjacency="r" deleteListLocation="C:\My  
Documents\dave\project\support\deleteList.txt" />
```

## Step 4: Run the script

Navigate to the directory containing AutoMap3. At the command prompt type `am3script project.aos`. Check for your output files in the directory designated in the `.aos` file

## Setting up and Running a Script

### Description

Running the script requires the use of the Command Line Prompt. This is found in the Start menu. It's exact location may be different depending on the setup of your particular computer. It is normally found in the "All Programs" option in the "Accessories" directory.

### Step 1. Create WorkSpace for input & output files

Navigate to your workspace and create a project directory. Inside this directory create an input and output directory.

```
C:\My Documents\dave\project  
C:\My Documents\dave\project\input  
C:\My Documents\dave\project\output  
C:\My Documents\dave\project\support
```

### Step 2. Place your text files in the input directory

Copy all your text files into the `C:\My Documents\dave\project\input` directory.

### Step 4. Place your work files in a directory

Place any Delete Lists and Thesauri in the `C:\My Documents\dave\project\support` directory.

## Step 5. Make a new .aos file

Make a copy of the standard .aos file and open it in a text editor. Tell AutoMap where your input files are and where you want the output written. Under the <AutoMap> tag is textDirectory (where you placed your text files) and tempWorkspace (where you want AutoMap to write your output files. This config file is setup to apply a thesarus, apply a delete list, and produce concept lists.

## Step 6. Determine the Preprocessing functions to use

Review the list of AutoMap tags to determine which script tags will be necessary. Insert those tags into your new .aos file in the proper location. Set the parameters for each function you are using. Some functions do not require any additional parameters while others require to tell AutoMap the type of processing you want.

```
<?xml version="1.0" encoding="UTF-8"?>
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="" />
</Settings>
<Utilities>

<PreProcessing>
<Generalization thesauriLocation="C:\My
Documents\dave\project\thesauri.csv"
useThesauriContentOnly="y" />
<DeleteList adjacency="r" deleteListLocation="C:\My
Documents\dave\project\deleteList.txt" saveTexts="y"/>
</PreProcessing>

<Processing>
<ConceptList />
<UnionConceptList />
</Processing>

<PostProcessing>
</PostProcessing>
```

```
</Utilities>  
</Script>
```

Save this new .aos file in the same directory as the AM3Script file. If the file is not in the same directory then AM3 Script will fail.

## Step 7. Open a Command Window

From the Start Menu open a Command Run Window. By default this is in the Accessories folder but may be in a different location on your machine. Navigate to the location which contains AM3Script.

## Step 8. Run the Script file

Navigate to the directory containing AutoMap. At the prompt type  
am3script project.aos



### Description

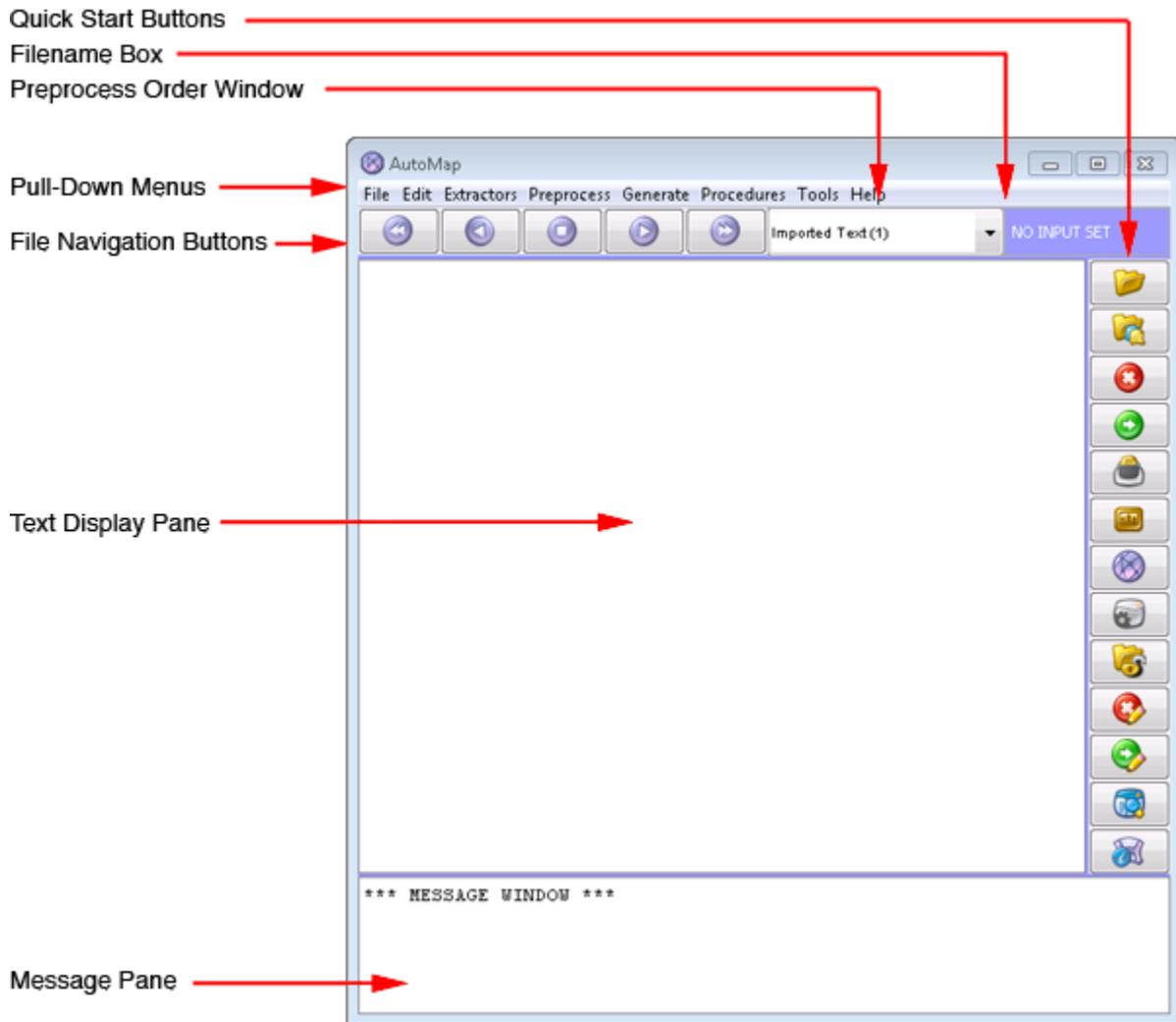
This section contains general information about the GUI.



### Description

The GUI (Graphical User Interface) contains access to AutoMap's features via the menu items and shortcut buttons.

## The GUI



## The Pull Down Menu

### File

Used for loading and saving text files. Can also extract text from other sources (PDFs and the Web).

### Edit

Allows the user to change the font of the **Display Window**

### Extractors

The File Menu contains functions whose main purpose is working with the files themselves. These functions do not perform any processing or generate any output. They work with raw files to help prepare them for use in AutoMap.

## Preprocess

Contains all the preprocessing functions used on your text before generating output. These functions work on original text files only.

## Generate

Generates output from preprocessed files. The output is directly related to the work done with preprocessing tools.

## Tools

External Tools used by AutoMap to View and Edit output. Tools include Concept List and Semantic List viewers and Delete List and Thesaurus editors.

## Help

The Help file and about AutoMap.

## File Navigation Buttons

Used to display the files in the main window. The buttons contain from left to right: **First** (lowest numbered file), **Previous** (previous file in sequence), **Goto** (Enter number of specific file), **Next** (next file in sequence), and **Last** (last numbered file)

## Keyboard Shortcuts

You can navigate through your loaded files using the cursor keys.

### Arrow-Left

Go to the Previous Text File.

### Arrow-Right

Go to the Next Text File.

### Arrow-Up

Go to the First Text File

**Arrow-Down**

Go to the Last Text File

## Preprocess Order Window

Contains a running list of the preprocesses performed on the files in the order they were performed. These can be undone one process at a time with the Undo command starting with the last process done. They can not be undone out of order.

## Filename Box

Displays the name of the currently active file along with it's ordinal number and the total number of files loaded. Using the File Navigation Buttons will change this and as well as the text displayed in the window. This also displays the total number of files loaded and the order number of the currently displayed text.

## Text Display Window

Displays the text for the currently selected file. The name of this file can be found in the Filename Box.

## Message Window

Area where AutoMap display the actions taken as well errors encountered. This area is also a place the user can insert notes about the current session. This can be helpful for later reference. You can copy text from the display window or type notes directly.

## Quick Launch Buttons

The Quick Start Buttons contain the most frequently used tools.

**NOTE :** *More detailed information about the various functions can be found in the Content and Task sections.*

## Description

### Show MetaNetwork Text Tagging

Creates a MetaNetwork (Carley, 2002) List for each loaded file based on a selected MetwNetwork Thesaurus. AutoMap will ask you to specify a target directory for the lists it creates. Will tag any concept found in the MetaNetwork Thesaurus. All others are tagged as **UNKNOWN**.

### Show Part of Speech Tagging

Parts of Speech assigns a single best **Part of Speech**, such as noun, verb, or preposition, to **every word in a text**. While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words can match multiple tags, depending on the context that they appear in.

...  
Roman, JJ  
citizens, NNS  
wandering, VBG  
the, DT  
...

**NOTE :** When finished reviewing the information you should **Undo** this item or else any further processing will be done using the results obtained. You mostly want to see the results without have AutoMap continue processing with these results.

### Filter Words

Enter a word or words you want to filter your Imported Text through. Upon finishing AutoMap will display only those words for each text.

**NOTE :** When finished reviewing the information you should **Undo** this item or else any further processing will be done using the results obtained. You mostly want to see the results without have AutoMap continue processing with these results.

## Hide Known Entries

This routine will prompt for a master thesauri and then hide known thesauri entries in the main text window.

**NOTE :** When finished reviewing the information you should **Undo** this item or else any further processing will be done using the results obtained. You mostly want to see the results without have AutoMap continue processing with these results.

---

[Preference Menu](#) : Where you can change various parameters affecting how AutoMap functions.



## Description

The following are short descriptions of the functions from Generate Pull Down menu. These functions generate output from preprocessed files.

When you run any of the generate functions AutoMap will create a new folder for the results. The folder will begin with the preprocess function end with a number (e.g. MetaNetwork1, MetaNetwork2...). AutoMap will find the last number in the series and increment the number by one. If no folder exists then AutoMap will create a new folder starting with 1.

## Text Properties

Outputs information regarding the currently loaded files. AutoMap writes one file for each file currently loaded containing.

Number of Characters, 14369  
Number of Clauses, 325  
Number of Sentences, 167  
Number of Words, 2451

See [Content => Text Properties](#) for more information.

## Named Entities

Named-Entity Recognition allows you to retrieve proper names, numerals, and abbreviations from texts.

See [Content => Named Entity](#) for more information.

## Data Extraction

The Feature Selection creates a list of concepts of money, dates, phone numbers and times.

See [Content => Feature Extraction](#) for more information.

---

	<b>Generate-Parts Of Speech</b>
---	---------------------------------

## Part of Speech Tagging

Parts of Speech assigns a single best **Part of Speech**, such as noun, verb, or preposition, to **every word in a text**. While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words can match multiple tags, depending on the context that they appear in.

AutoMap will ask you how you want to save your files. First Automap will ask if you want **Standard** (the entire list of tags) or **Aggregation** (a

consolidated list) Parts of Speech tagging. Second you will be asked to save them in the **CSV** or **TXT** format.

...  
Roman, JJ  
citizens, NNS  
wandering, VBG  
the, DT  
...

See [Content => Parts of Speech](#) for more information.

## POS Attribute File

Similar to the above function but if there are multiple occurrences of the same concept it will assign **the best possible Part of Speech** to a concept.

...  
battlefield, NN  
volumnius, PRP  
benefit, NN  
angrily, CD  
...

## Verb Extraction

Compiles of list of all actions (verbs) in the specified file.

## Noun Extraction

Compiles of list of all nouns and in the specified file.

	<b>Generate-Concept Lists</b>
---	-------------------------------

## Concept List (Per Text)

Generates a Concept List for all loaded files. The list contains a concept's frequency (number of times it occurred in a file), relative frequency (a

concept's frequency in relationship to the total number of concepts). A Concept List can be refined using other functions such as a Delete List (to remove unnecessary concepts) and Generalization Thesaurus (to combine n-grams into single concepts).

concept	pos	frequency	relative frequency within text	gram type	number of texts	meta
Antony	NNP VBN	4	0.14814815	single	1	UNKNOWN
Brutus	EX IN JJ NN NNP PRP VBN	16	0.5925926	single	1	UNKNOWN
Caesar	DT NNP VB VBN	27	1.0	single	1	UNKNOWN

See [Content => Concept List](#) for more information.

## Concept List (Union Only)

The Union Concept List differs from the Concept List in that it considers concepts across all texts currently loaded, rather than only the currently selected text file. The Union Concept List is helpful in finding frequently occurring concepts, and after review, can be determined as concepts that can be added to the Delete List.

See [Content => Union Concept List](#) for more information.

## Concept List with MetaNetwork (Carley, 2002) Tags

Creates a Concept List which lists a MetaNetwork category if applicable.

## Concept Network DyNetML (Per Text)

Creates a separate DyNetML file of concepts for each text file loaded. These files are directly usable in ORA.

## Concept Network DyNetML (Union Only)

Creates one DyNetML file of the concepts in all text files loaded. This is a union file of all concepts. This file is directly usable in ORA.

**NOTE :** Both Concept Network functions create DyNetML files with one NodeClass and no Networks. Making the connections is up to you after importing the file into ORA.

**NOTE :** Leading and trailing hyphens are removed before generating Concept Lists and Semantic Lists but hyphens in the middle of two words are not (e.g. **because-- something** removes the double hyphens but in the concept **t-shirt** the hyphen would not be removed).

## Keywords in Context

A list will be created so every concept in a file along with the concepts which both precede it and following it.

```
concept , left , right
Two , , tribunes
tribunes , Two , Flavius
Flavius , tribunes , and
and , Flavius , Murellus
...
```

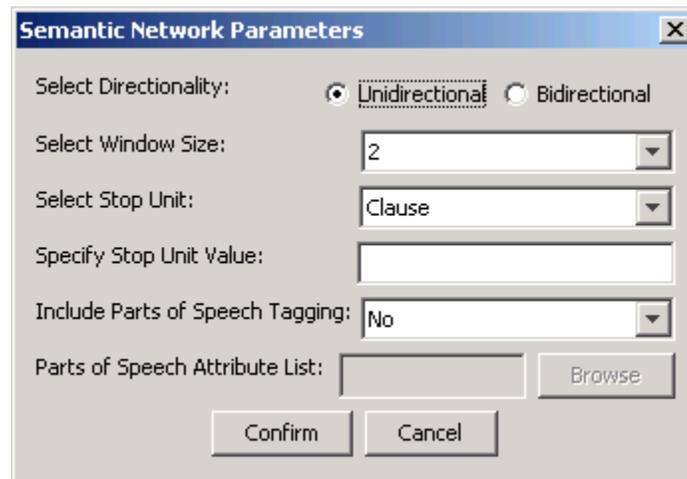
**NOTE :** The first entry **Two , , tribunes** contains a blank entry for **left** as it's the first word in the text and has nothing to the left. A similar entry will be found at the end with a blank in the column **right**.

	<b>Generate-Semantic Networks</b>
---	-----------------------------------

## Semantic Network DyNetML (Per Text)

Semantic networks are knowledge representation schemes involving nodes and links between nodes. It is a way of representing relationships between concepts. The nodes represent concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph. Semantic Networks created can be displayed in ORA.

**NOTE : Use All Words As Window Size :** When generating a Meta Networks, Semantic Networks or Semantic Lists, the user now has the option of specifying whether or not they want to use all words in a sentence as the window size. Using this option will automatically set the stop unit to "sentence".



## Semantic Network DyNetML (Union Only)

Creates union file of all DyNetML files in one directory. Before running this make sure that only the DyNetML files you want to union reside in the directory chosen.

<p

See [>Content => Semantic Network](#) for more information.

## Semantic (Co-Reference) List

Semantic Lists contain pairs of concepts found in an individual file and their frequency in the chosen text file(s).

See [Content => Semantic List](#) for more information.

**NOTE** : Leading and trailing hyphens are removed before generating Concept Lists and Semantic Lists but hyphens in the middle of two words are not (e.g. **because-- something** removes the double hyphens but in the concept **t-shirt** the hyphen would not be removed).

## Noun Phrase to Modifiers List

Creates a .csv file containing what AutoMap believes to be noun phrases using the format **concept, concept, frequency, metaOntology, metaName**.

## Concept to Source List

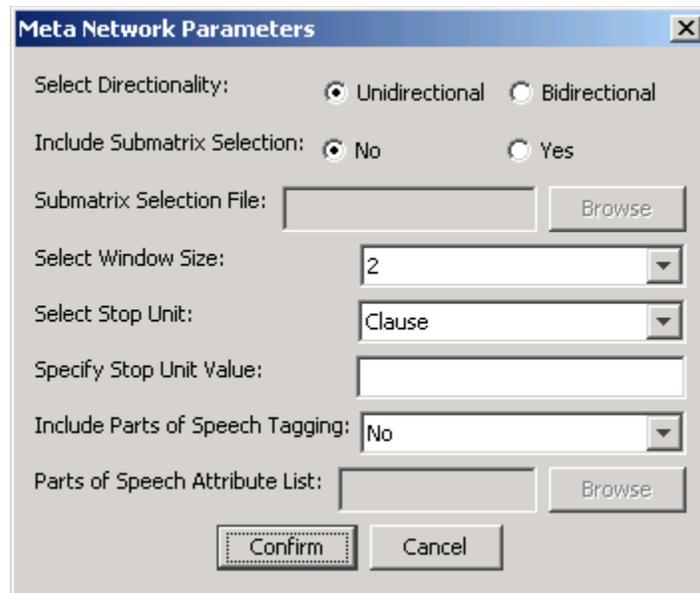
Creates a csv file containing a list of concepts and their origin file and frequency, **concept, source, frequency, metaOntology, metaName**.

## Modal Rule List



## MetaNetwork DyNetML (Per Text)

Assigns MetaNetwork (Carley, 2002) categories to the concepts in a file. This is used to create a DyNetML file used in ORA.



**Select Directionality** sets whether AutoMap will search only forward in the text or will perform a search both forward and backward in the text. **Select Window Size** sets the farthest distance from a word to another for a possible connection. **Select Stop Unit** contains Word, Clause, Sentence, Paragraph, or All.

**NOTE :** The panel contains defaults for all parameters except the **Specify Stop Unit Value**

**NOTE : Use All Words As Window Size :** When generating a Meta Networks, Semantic Networks or Semantic Lists, the user now has the option of specifying whether or not they want to use all words in a sentence as the window size. Using this option will automatically set the stop unit to "sentence".

## MetaNetwork DyNetML (Union Only)

Creates union file of all MetaNetwork (Carley, 2002) DyNetML files in one directory. Before running this make sure that only the MetaNetwork (Carley, 2002) files you want to union reside in the directory chosen.

**NOTE :** The Union type is a sum type.

## MetaNetwork Text Tagging

Tags any entry found in the text files which exists in the selected General Thesaurus using the format **concept/part of speech**.

## Suggested Name Thesauri

Creates a file with the following attributes: conceptFrom, conceptTo, frequency, relative\_frequency-across\_texts, relative\_percentage-across\_texts, number\_of\_texts, metaOntology, metaName.

And example taken from the unprocessed Julius Caesar files:

conceptFrom	conceptTo	frequency	relative frequency across texts	relative percentage across texts	number of texts	metaOntology	metaName
Julius Caesar	Julius_Caesar	1	0.0192	0.0030	1	agent	
Brutus	Brutus	4	0.0769	0.0122	2	agent	
Cassius	Cassius	13	0.2500	0.0396	4	agent	

## Suggested Uncategorized Thesauri

conceptTo	pos	metaOntology	conceptFrom
That	DT	knowledge	
angrily	CD		

parade	NN	resource	
wrongly	RB	knowledge	



## Generate-Thesaurus Suggestion

## Suggest MetaNetwork Thesauri (Unigrams Only)

Automatically estimates mapping from text words from the highest level of pre-processing to the categories contained in the Meta-Network. The technology used is a probabilistic model based on a conditional random fields estimation. Suggested thesaurus is a starting point.

```

1In, resource
1On, resource
Cicero, agent
sons, agent
streets, location
Brutuss, agent
women, agent
prisoner, agent
4Portia, resource
masses, agent
.....

```

A MetaNetwork (Carley, 2002) Thesaurus associates concepts with the following metanetwork (Carley, 2002) categories: **Agent, Knowledge, Resource, Task, Event, Organization, Location, Action, Role, Attribute, and a user-defined categories.**

**NOTE :** *The more the text is modified the less accurate the CRF generator will be.*

See [Content => MetaNetwork](#) for more information.

**NOTE :** *The following five functions create a Suggested MetaNetwork Thesaurus along with cleaned files that use the newly created Thesaurus.*

**OneMode :** *Assigns a single metaOntology to each entry*

**MultiMode :** *Assigns all matching metaOntologies to each entry*

## **Suggest Entity Thesaurus (OneMode, includes multiword expressions)**

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

## **Suggest Entity Thesaurus (MultiMode, includes multiword expressions)**

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

## **Suggest MetaNetwork Thesaurus, Categories and Specificity, recommended default (Multimode, includes multiword expressions)**

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, metaType, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

## Suggest MetaNetwork Thesaurus, Categories and Subtype (Multimode, includes multiword expressions)

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, metaType, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

## Suggest MetaNetwork Thesaurus, Categories, Specificit, and Subtype (Multimode, includes multiword expressions)

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, metaType, metaname, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

## Decision Support Wizard

A simple chart to assist you in determining the correct method to Thesaurus Creation.



## BiGrams

BiGrams are two adjacent concepts in the same sentence. If a Delete List is run previous to detecting bi-grams then the concepts in the Delete List are ignored. Multiple Delete Lists can be used with a set of files.

**NOTE :** *The two concepts of a bigram can not cross a sentence or paragraph boundary*

See [Content => BiGrams](#) for more information.

## Context-Sensitive Stemming Thesauri

Takes concepts down to their base forms. It makes a thesauri for users to evaluate and run.

- It depluralizes nouns, such as "boys" to "boy".
- It detenses verbs, such as "ran" to "run".

## Non-Context Stemming Thesauri

Creates a thesaurus with the information **conceptFrom, conceptTo, metaOntology, metaName, frequency, POS**. Will stem entries to their bases. [ i.e. **believing > believe, gives > give, and enraged > enrage** ]

### Identify Possible Acronyms

Given a directory with text files in it, Identify Possible Acronyms will scan through the text files of the given directory and compile a list of possible acronyms. It identifies acronyms as a series of letters in all upper-case, so not all acronyms will be completely accurate; hence why it is a list of possible acronyms. The program takes an input directory and an output directory, and creates a single CSV named >s[pan>possibleAcronyms.csv. It has two column headers: one for the acronym, and the other for the frequency in which acronym appears across all text files. So an example file would contain such:

```
CONCEPT , FREQUENCY  
US , 3  
WHO , 5  
USA , 10
```

### Suggest NGram List

Computes a list of possible NGrams from a given set of input text files. It creates a list of **bigrams, trigrams, quadgrams and quintgrams** and then unions those lists together. Invalid ngrams specified by a set of rules the program follows are removed from this union list before it is written out to the file specified by the user.

### Positive Thesaurus

A Positive Thesaurus takes every concept in the text and defines it as itself. This can be used as the start in building a Generalization Thesaurus.

**NOTE** : This function is **case specific** meaning if the concepts **He** and **he** both appear in the text they will both appear in the newly created thesaurus.

*fido.txt*

John has a dog named Fido

### **Positive Thesaurus**

John, John

has, has

a, a

dog, dog

named, named

Fido, Fido

	<b>Generate - Networks Uncovered By Bayesian Inference (NUBBI)</b>
--	--

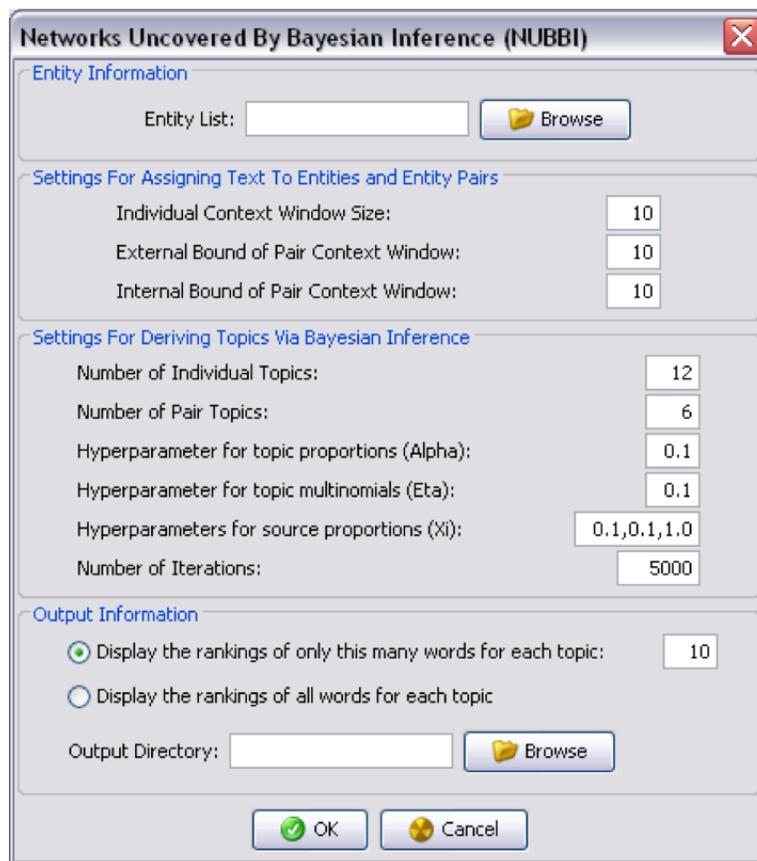
## What is NUBBI?

Networks Uncovered By **Bayesian Inference (NUBBI)** is a topic modeling scheme introduced by Chang et al. in 2009. NUBBI is an exploratory method for describing characteristic relationships between entities in a document. When run, it uses the text that often occurs near entities in order to generate a set of characteristic "topics" that describe both entities and pairs of entities.

These pair topics can be considered networks of particular relationships with entities. Where links in a Semantic Network can be construed as indicating that "Entity A and Entity B are mentioned within N words of each other", a link a Pair Topic network can be construed as meaning that "Entity A and Entity B are associated with a particular distribution of vocabulary words with strength X." If the pair topic favors words relating to sports, the tie strength characterizes how strongly they are associated with sports. The key element is the distribution of words associated with the topic....

This help document describes both how to run NUBBI in AutoMap and how the algorithm works in AutoMap. You can review the papers in the reference section for more information about the theoretical grounding and validation of the algorithm.

## NUBBI Interface



The screenshot shows a dialog box titled "Networks Uncovered By Bayesian Inference (NUBBI)". It is divided into four sections:

- Entity Information:** Contains an "Entity List:" text box and a "Browse" button.
- Settings For Assigning Text To Entities and Entity Pairs:** Contains three spinners: "Individual Context Window Size" (10), "External Bound of Pair Context Window:" (10), and "Internal Bound of Pair Context Window:" (10).
- Settings For Deriving Topics Via Bayesian Inference:** Contains five spinners: "Number of Individual Topics:" (12), "Number of Pair Topics:" (6), "Hyperparameter for topic proportions (Alpha):" (0.1), "Hyperparameter for topic multinomials (Eta):" (0.1), and "Hyperparameters for source proportions (Xi):" (0.1,0.1,1.0). It also has a "Number of Iterations:" spinner set to 5000.
- Output Information:** Contains two radio buttons: "Display the rankings of only this many words for each topic:" (selected) and "Display the rankings of all words for each topic:". The selected option has a spinner set to 10. Below is an "Output Directory:" text box and a "Browse" button.

At the bottom are "OK" and "Cancel" buttons.

The user must set several different parameters when using NUBBI. All have default values except for two in order to increase usability. Increasing the numbers of topics and number of iterations are the easiest things to change in order to improve your exploration with the algorithm

### Entity List:

The Entity List should be a list of the different entities that you want to situate in the metanetwork. This file must be in Master Thesaurus format. NUBBI only uses the conceptTo and metaOntology columns; your document should be generalized beforehand. Entities without an entry in metaOntology will be assumed to be Agents.

### **Individual Context Window Size:**

The number of words (up to and including the threshold word) on either side of a mention of an entity will be considered part of an entity context.

### **External Bound of Pair Context Window:**

The external bound on pair context size. Given that two entities are in the same context, this bound sets the number of threshold words (up to and including the threshold word) that should be considered part of the pair context.

### **Internal Bound of Pair Context Window:**

The maximum number of words (up to and including the threshold word) permitted between two entities that are part of the same pair context.

### **Number of Individual topics:**

The number of individual topics to use in the model.

### **Number of Pair Topics:**

The number of pair topics to use in the model.

### **Hyperparameter for topic proportions (Alpha):**

A hyperparameter that tunes the relative weight placed on the assignment of a word to a particular topic, given its assignment to a particular source (see below). Increasing alpha decreases the weight.

### **Hyperparameter for topic multinomials (Eta):**

A hyperparameter that tunes the relative weight placed on the assignment of a word to a particular topic, given the number of times any word has been assigned to that topic (see below). Increasing eta decreases the weight.

### **Hyperparameters for source proportions (Xi):**

A group of three hyperparameters that, for a pair context, scales the weight of assignment to the first member of a pair, second member of a pair, or the pair itself. Because of the sequential nature of the stick-breaking construction (see below), equal weights on the first member of a pair will favor it over the second member.

### **Number of Iterations:**

The number of iterations of the algorithm to run.

### **Display the rankings of only this many words for each topic:**

If you specify this option, you must also choose the top N words whose rankings will be displayed. In the final network output, the N words most closely associated with each topic will be listed.

### **Display the rankings of all words for each topic:**

If you specify this option, in the final network output all words in the vocabulary will be listed in ranked order for each topic.

### **Output Directory:**

AutoMap will create a folder in the directory you specify called "nubbiNetworks", followed by a numerical suffix. This folder will contain all output related to NUBBI. (See below.)

## ***The NUBBI Algorithm***

The NUBBI Algorithm consists of two distinct components: the parsing of a set of text into words that apply to individual entities and pairs of entities, and the probabilistic allocation of the words in these contexts to pair- and entity topics.

## ***Text splitting***

NUbbi reviews all of the texts loaded into AutoMap. When an entity is found with no other entity occurring within the internal bound of the pair context window, the surrounding text is copied into an entity context. When a pair of entities are found together, the surrounding text is stored in a pair context. If multiple entities occur within a single pair context, the text is copied and included in the pair contexts for each pair of entities.

## ***Topic Generation***

After the texts have been split into pair- and entity-contexts, NUBBI repeatedly iterates through the texts, assigning all occurrences of a particular word within a context to an entity and topic. The general form of an iteration is as follows:

For a particular context:

For all of the occurrences of a particular word in that context:

- Remove the weight of the word from the probability calculation. (Discount all of its assignments to different weights.)
- Calculate the probability of assigning the word to each possible topic. (If an entity, this means just including the entity topics. If a pair of entities, this means two sets of entity topics and the pair topics.)

Given that...

- $k$  is a topic drawn from the set of pair or entity topics

- $d$  is the current pair- or entity-context
- $w$  is the current word in  $d$
- $s$  is the current source in  $d$
- the probability of assignment to topic  $k$  is:

$$\frac{\alpha + \text{words assigned to } s \text{ and } k}{\alpha \cdot \text{topics} + \text{words assigned to } s}$$

$$\cdot \frac{\eta + \text{times } w \text{ has been assigned to } k}{\eta \cdot \text{vocabulary} + \text{times words have been assigned to } k}$$

$$\cdot \text{words of } d \text{ assigned to source } 1, 2, \text{ or pair} \cdot \xi_{\text{source } 1,2,\text{or pair}}$$

- Draw a random number, and use it to reassign the word. In accordance with the "stick-breaking" construction of the Dirichlet distribution we check each topic in sequence, not in random order, and similarly begin with the first member of the pair, then the second member of the pair, and then the pair itself.

## Output

NUBBI produces several forms of output:

- ***nubbiNetworks.xml*** - A collection of DyNetML networks and node classes.
  - ***Node classes***
    - There will be a node class for the specified class of each observed entity (as dictated in the entity list thesaurus)
    - A node class for all of the different topics, called "Topics", and of type "Resource". Each node has a set of ranked words listed as

attributes. From this set of attributes, you can develop an appropriate contextual label for each topic.

- A node class for all of the different concepts, called "Concepts" and of type "Knowledge".

▪ **Networks**

- $\langle \text{Nodeclass} \rangle \times \text{Topic}$  networks for the observed node classes in the entity list. The tie strength is the total number of words associated with each entity that are also associated with that topic.
- Pair topic networks for every pair topic and combination of node classes observed to co-occur. The tie strength is the total number of words associated with the pair and the particular topic.
- A Concept  $\times$  Topic linking every concept with every topic, with tie strength equal to the number of times each word has been associated with a particular topic, regardless of context.

**The "alphabetDir" folder**

- **alphabet.txt** - A complete list of all the vocabulary read in the file, separated by spaces.
- **entities.txt** - A complete list of all the entities used in the file, separated by new lines. Note that this file may be a subset of the entities in the initial entity list, if not all entities are observed.

**The "contextDir" folder**

- A set of files, labeled **0.txt** through **(N-1).txt**, where N is the number of observed entities. Each file contains counts of all of the vocabulary words observed and the frequency with which they occur in the same context as the entity. All of words are coded as 0-indexed numbers keyed to the alphabet.txt file. The first number in each line is the coded word, while the second is the number of times it occurred.

### ***The "pairContextDir" folder***

- A set of files, similar to those in **contextDir**, but containing counts of the vocabulary words observed and the frequency with which they occur in a particular pair context.

### ***The "matrices" folder***

- ***document\_source\_sums.csv*** - A matrix of documents x source assignment (0, 1, or the complete pair), where each cell contains the number of words in the document assigned to a particular source.
- ***document\_sums.csv*** - A document x topic matrix, where each cell contains the number of words in the document assigned to a particular topic.
- ***wordsByTopics.csv*** - A matrix of vocabulary words x topics, where each cell contains the number of times a particular word was assigned to a topic.

## ***References***

---

[1] J. Chang, J. Boyd-Graber, and D. M. Blei, "Connections between the lines: augmenting social networks with text," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France, 2009, pp. 169–178.

<http://dl.acm.org/citation.cfm?id=1557044>

[2] J. Chang, "Unconverging, Understanding, and Predicting Links," Princeton University, Princeton, New Jersey, 2011.

<http://arks.princeton.edu/ark:/88435/dsp010r9673747>



## Description

This group of functions work on files other than the currently loaded text files.

### Validate Script

Determines whether a script is valid to run in AutoMap.

### Basic Model Wizard



**IMPORTANT NOTE :** It is necessary to make sure the headers in a Master Thesaurus contain the proper headers before using them.

conceptFrom	conceptTo	metaOntology	metaName
-------------	-----------	--------------	----------

They do not necessarily have to be in that order but they need to be those exact names.

### Master Thesauri Merge

Click the Original Master Thesauri [**Browse**] button and select a file to change. Click the Change Thesauri [**Browse**] button to select a second Thesauri file. Underneath use the radio buttons to select the type of File this is. Click the Output Master Thesauri Directory [**Browse**] button and navigate to the location to save the new file.

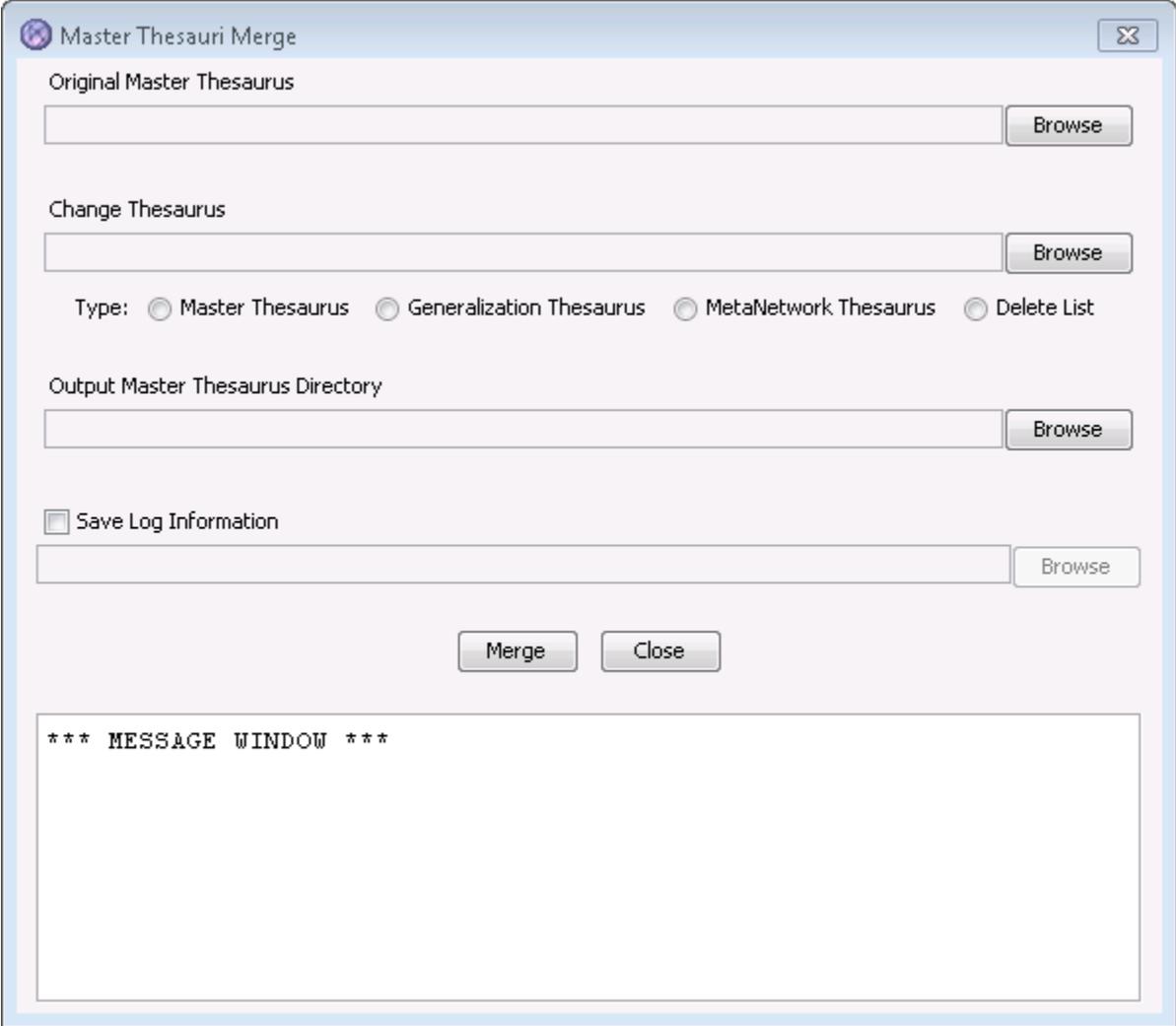
### Convert UTF Entries to ASCII Entries

Converts the **unreadable** characters into their readable equivalent. This works on the thesauri files. If there is no equivalent for a character on the line, it is written out to a rejects file. You will be asked for [1] the file to convert; [2] the name of the file to write converted characters; and [3] the name of the file for leftover characters which can not be converted.

**NOTE :** All three files require the **.csv** extension.

**NOTE :** If a check mark is placed in the **Save Log Information** you can use the Click the [**Browse**] button to select a location to save this file.

When finished click [**Merge**] to create the newly merged file.



The screenshot shows a dialog box titled "Master Thesauri Merge". It contains several input fields and buttons:

- Original Master Thesaurus:** A text input field with a "Browse" button to its right.
- Change Thesaurus:** A text input field with a "Browse" button to its right.
- Type:** A group of four radio buttons: "Master Thesaurus", "Generalization Thesaurus", "MetaNetwork Thesaurus", and "Delete List".
- Output Master Thesaurus Directory:** A text input field with a "Browse" button to its right.
- Save Log Information:** A checkbox that is currently unchecked, followed by a text input field and a "Browse" button.
- Buttons:** "Merge" and "Close" buttons are located at the bottom center.
- Message Window:** A large text area at the bottom containing the text "\*\*\* MESSAGE WINDOW \*\*\*".

## Convert Review to Master

### Approximate Closeness of Names

This procedure takes as input a master thesaurus and a threshold percentage of how similar you want names to be. What is output is a master thesaurus with similar names being generalized to another similar name. For example, given the following input:

```
conceptFrom,conceptTo,metaOntology,metaName
Dan,Dan,agent,casos
Dave,Dave,agent,casos
Frank,Frank,agent,casos
France,France,location,country
Mike,Mike,agent,casos
Pike,Pike,resource,weapon
```

The output we get when using a word distance threshold of 80% is:

```
"conceptFrom","conceptTo","metaOntology","metaName"
"Frank","France","agent",""
"Mike","Pike","agent",""
```

**NOTE :** A **Word Distance Threshold** of 95% gives a good balance.

### Extract Thesaurus Attributes

Take apart an input master thesaurus and break it down into multiple files based on an entry's ontological value. For example, all of the agent types in a sample input thesaurus will show up in the **inputName\_agents.csv** file generated by this procedure. The same can be said for organizations, locations, tasks, resources, and so on. A meta thesaurus, a delete list and a generalization thesaurus are also created from this process.

## Identify Bad Characters in Thesaurus Entries

This will go through a thesaurus and identify each potentially problematic character on each line. The output will list the line number of the thesaurus first, and then all of the problem characters found, then each entry where that problem character occurs. Here is the list of characters we identify as bad:

```
' ~ = [ ] \ \ ; ` . / ! @ # $ % ^ & * ( ) + { } | : < >
? \t
```

## Identify Thesauri Noncategorized Entries

Takes as an input a Master Thesauri and will display in the message window all entries which have no metaOntology listed for it. This list will appear in the Message Window. The information in this window can be to a file via **File > Save Message Window Log** for use in other programs.

***NOTE :** Does not work on Generalization Thsauri. The must first be converted to the Master Thesauri format.*

## Derole Thesauri Entries

Takes as input a thesaurus and outputs a thesaurus, both in master format. Will attempt to find **roles** in both the **conceptFrom and conceptTo** columns of the thesaurus and add de-roled terms to the thesaurus. An attribute file is also output from the program, that contains a list of what roles are mapped to which concepts.

*The entry **President Barack Obama** would add two concepts to the thesaurus: 1) **Barack Obama** and 2) **President Barack Obama**.*

## Apply Thesauri as Delete List

Takes three different Master Thesauri as arguments: an input thesaurus, an output thesaurus and a delete thesaurus. The Delete Thesaurus is treated as a Delete List and is applied to the Input Thesaurus.

**NOTE :** *The Master Format is required for all arguments.*

## **Apply Ontology Rules**

Takes an Input Thesaurus and outputs a Thesaurus with modified Meta Ontology values. The program reads in a encrypted rules file, which contains rules to reclassify a concept's meta ontology value based on patterns.

**Example :** A concept that contains **Press Release** at the end of it would be reclassified to a task. The current rules file is encrypted distributed within the installers.

## **Remove Noise Patterns**

Takes an Input Thesaurus and outputs a Thesaurus with special patterns stripped out of the list of concepts.

**NOTE :** *Examples would be: letter-\_letter or -\_letter or \_letter.*

## **Separate Number Terms from Thesauri**

Takes an Input Thesaurus and outputs two different Thesauri. The first is a thesaurus with all number concepts stripped out of it, except number concepts that are potentially locations. The second thesaurus is a thesaurus of only the number concepts that have been removed from the input thesaurus.

## **Resolve Names**

This is a program that takes a master thesaurus as input and outputs a master thesaurus. The program will scan through the conceptFrom column of the input thesaurus and find entries that have a meta ontology value of agent. The program will then compile a list of possible names to resolve to, only storing the longest possible term for each name. Lastly, the program will scan through the list of agents in the thesaurus once more and -- if the entire term is a part of the full name listed -- the program will set that term's conceptTo column as the full name.

### This :

```
Mark Godwin,Mark_Godwin,agent ,person  
Mark,Mark,agent ,  
Godwin,Godwin,agent ,
```

### Will be resolved to :

```
Mark Godwin,Mark_Godwin,agent ,person  
Mark,Mark_Godwin,agent ,  
Godwin,Mark_Godwin,agent ,
```

**NOTE :** This feature has been implemented into the script, the AutoMap GUI and the Script Runner GUI. It has also become a part of the deletion process and will automatically be run when NameThesaurusRevision is called.

## Remove Leading Article

Takes a Master Thesaurus as input and outputs a Master Thesaurus. It will scan through the **conceptTo** column of the input thesaurus and find entries that begin with either **a, an or the**. Those prefixes are then removed.

### Start with :

```
The John Smith  
Corporation,The_John_Smith_Corporation,organization,
```

### Will change to :

```
The John Smith  
Corporation,John_Smith_Corporation,organization,
```

## Split Compound Thesauris Entries

Takes a Master Thesaurus as input and outputs a Master Thesaurus. Scan through the **conceptFrom** column of the input thesuarus and find entries that contain **and, or, and the bullet character (\u2022)**. It then takes that concept apart and adds each separated concept to the thesaurus as a term, with its meta ontology value being derived from the compound concept.

**NOTE :** The only exception to this is if the program encounters an organization with **and** in it. If there is one **and**, then the concept is left together. Otherwise, it is separated.

**Example :**

```
Blue Cross and Blue  
Shield,Blue_Cross_and_Blue_Shield,organization,  
Andy and Brian and Charlie and Donna and Ed and  
Frank,Andy_and_Brian_and_Charlie_and_Donna_and_Ed_and_Frank,  
agent  
•_eggs_•_milk_•_bread_•_cinnamon_powder_•_cheese,resource,
```

**Will change to :**

```
Blue Cross and Blue  
Shield,Blue_Cross_and_Blue_Shield,organization  
Dan,Dan,agent,  
Mike,Mike,agent,  
Frank,Frank,agent,  
Dave,Dave,agent,  
Jessica,Jessica,agent,  
Bradley,Bradley,agent,  
eggs,eggs,resource  
milk,milk,resource  
cinnamon powder,cinnamon_powder,resource,  
cheese,cheese,resource,
```

## Remove Date Entries

Looks in Concept List and any entry that matches the date format is removed.

## Revise Name Thesaurus

Takes a Master Thesaurus as input and outputs a master thesaurus. Scans through the **conceptFrom column** of the input thesaurus and find entries that have a meta ontology value of agent. Then it compiles a list of possible names to resolve to, only storing the longest possible term for each name. Lastly, it will scan through the list of agents in the thesaurus once more. And if the entire term is a part of the full name listed the program will set that term's **conceptTo column** as the full name.

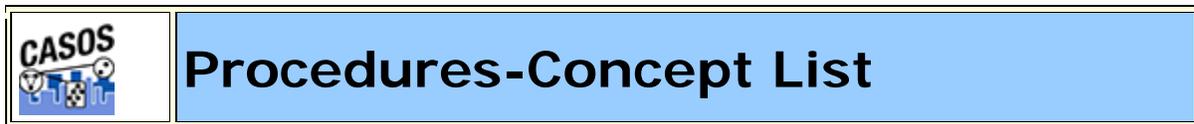
**Example :**

Mark Godwin,Mark\_Godwin,agent ,person  
Mark,Mark,agent ,  
Godwin,Godwin,agent ,

**Will resolve to :**

Mark Godwin,Mark\_Godwin,agent ,person  
Mark,Mark\_Godwin,agent ,  
Godwin,Mark\_Godwin,agent ,

## Split to Individual Ontologies



## Concept List Procedures

### Union Concept List Together

With this function you can join **any** concept lists into a Union Concept List file, even if they are from different textsets. Place all the concept lists you want to union into an empty directory. Then navigate this function to that directory. It will create a union of all the files in a newly created sub-directory called **union**.

### Concept List Trimmer

First you select **Trim by file percentage** or **Trim by frequency percentage**. AutoMap will ask for a Concept File to trim then a name for the new file. Next you will be asked for either a percentage or frequency to trim the file.

### Apply Delete List to Concept List

Allows you to choose a Delete List to apply to a selected Concept List

### Remove Integers from Concept List

Removes all numbers from a Concept List

## Convert Concept to Review



## Thesaurus Procedures

### Sort Thesaurus

In certain situations it is important to have your thesaurus sorted from longest to shortest before using it in the preprocess section. Entries with the most number of words are floated to the top of the list

## johnSmithDairyFarm.csv - Unsorted

```
John Smith,John_Smith  
cow,animal  
dairy farm,dairy_farm  
pig,animal  
The United States of America,the_USA  
chicken,animal  
Jane Doe,Jane_Doe
```

## johnSmithDairyFarm.csv - Sorted

```
The United States of America,the_USA  
John Smith,John_Smith  
dairy farm,dairy_farm  
Jane Doe,Jane_Doe  
cow,animal  
pig,animal  
chicken,animal
```

**The United States of America** with five words floats to the top. This is followed by the three entries **John Smith, dairy farm, and Jane Doe** each with two words. It finishes with three entries **cow, pig, and chicken** each with one word.

**NOTE :** *If your thesaurus has duplicate entries (e.g. "John,John\_Doe" and "John,John\_Smith") a warning will appear in the message window.*

**Warning:** *Duplicate entries found in thesaurus for "John".*

## Merge Generalization Thesaurus

Combine multiple Generalization Thesauri into one file. AutoMap allows you to select individual files from a directory.

**NOTE :** *When giving the new file a name remember to also add the .csv extension.*

**NOTE :** *If a concept exists in two thesauri but have different key\_concept values then both will be included in the merge.*

**NOTE :** *This procedure only supports the older file format. If you have a Master Format file then use the Master Thesaurus Merge.*

## Apply Stemming to Thesauri File

Takes a thesaurus file and creates new entries if a concept requires stemming. If multiple entries are stemmed to the same root and they have different key\_concepts then new entries will be added for each one.

*drive.csv*

drove,alpha  
driven,bravo

*Thesaurus after Stemming*

drove,alpha  
driven,bravo  
drive,alpha  
drive,bravo

## Apply a Delete List to a Thesaurus

You can use a Delete List to trim a Thesaurus.

---

## **Check Thesaurus for Missing Entries**

Checks a thesauri to find any where either line is blank.

## **Check Thesaurus for Duplicate Entries**

Checks if there are two entries referencing the same item. This is determined by the original concept.

## **Check Thesaurus for Circular Logic**

Sometimes, when creating a generalization thesauri, a concept is accidentally listed as both something to be replaced and something to replace another concept. For example:

```
United States,US  
cow,animal  
US,United_States_of_America
```

In this case, all instances of "United States" will first be changed to "US" and then to "United\_States\_of\_America". The Circular Logic Test alerts the user of this inefficiency.

## **Check Thesaurus for Conflicting Entries**

Will alert you if two or more Thesaurus entries are directed to replace the same concept.

---

The following four procedures convert files between formats as the names state.

## **Convert Master Thesauri to Generalization Thesaurus**

## Convert Generalization Thesauri to Master Thesaurus

## Convert Master Thesauri to Meta-Network Thesaurus

## Convert Meta-Network Thesauri to Master Thesaurus



## Delete List Procedures

### Apply Stemming to DeleteList File

Either the **K-Stem** or the **Porter** stemmer can be applied to a delete list, each with slightly different results.

*deleteListToStem.txt*

original list	K-Stem	Porter
drives	drives	drives
	drive	drive
wanted	wanted	wanted
	want	want
financial	financial	financial
		financi
motivation	motivation	motivation
		motiv

## Merge Delete Lists

Combine multiple Delete Lists into one file. AutoMap allows you to select individual files from a directory.

**NOTE :** *When giving the new file a name remember to also add the .txt extension.*

**NOTE :** *Wildcards are not supported when designating file names.*

**NOTE :** *This procedure only supports the older file format. If you have a Master Format file then use the Master Thesaurus Merge.*

## Convert Master Thesauri to Delete List

Takes a Delete List in the Master Thesauri format and converts it to a Standard Delete List.

### *Delete List - Master format*

```
"conceptFrom", "conceptTo", "metaOntology", "metaName"  
"a", "a", "#",  
"about", "about", "#",  
"actually", "actually", "#",  
"after", "after", "#",  
"all", "all", "#",
```

### *Delete List - Standard format*

```
a  
about  
actually  
after  
all
```

## Convert Delete List to Master Thesauri

Performs the complementary function of the preceding item.



## Procedures-DyNetML

### DyNetML Procedures

#### Union DyNetML

Will combine all DyNetML files within a single directory into a new DyNetML file.

#### Add Attributes

Used to add a attributes to a DyNetML file before importing them into ORA. The format of the attribute file is:

```
header row : NodeID, Attribute Name, . . . . , Attribute Name
data row : Node Name, Attribute value, . . . . , Attribute value
Additional rows or data
```

This will create an attribute column in the DyNetML underwhich all the values for identified nodes will be displayed.

**NOTE :** *If the DyNetML file does not contain a particular node\_ID then no information for that node\_ID will be added to the file.*

### Example

```
NodeID,color,shape
alpha,red,circle
beta,green,square
charlie,blue,triangle
```

#### Belief Enhancement

#### Relocate Source Location in DyNetML

Changes the source reference in a DyN etML file.

## Add Icon Reference to DyNetML

## Pairwise Union

Takes as input two DyNetML files which need to be in separate folders. It then creates a third DyNetML file which combines the nodes and links of the two source files.

**NOTE :** *The names of both source files needs to be identical.*

## Remove Semantic Networks

Removes the Semantic Network from a Meta-Network. This is done to reduce the size of the DyNetML if the analyst is only interested in the Meta-Network and has no use for the semantic net.

## Infer Kinship Links

Builds DyNetML files for all XML files located in the source directory. Results are based on frequency and proximity.



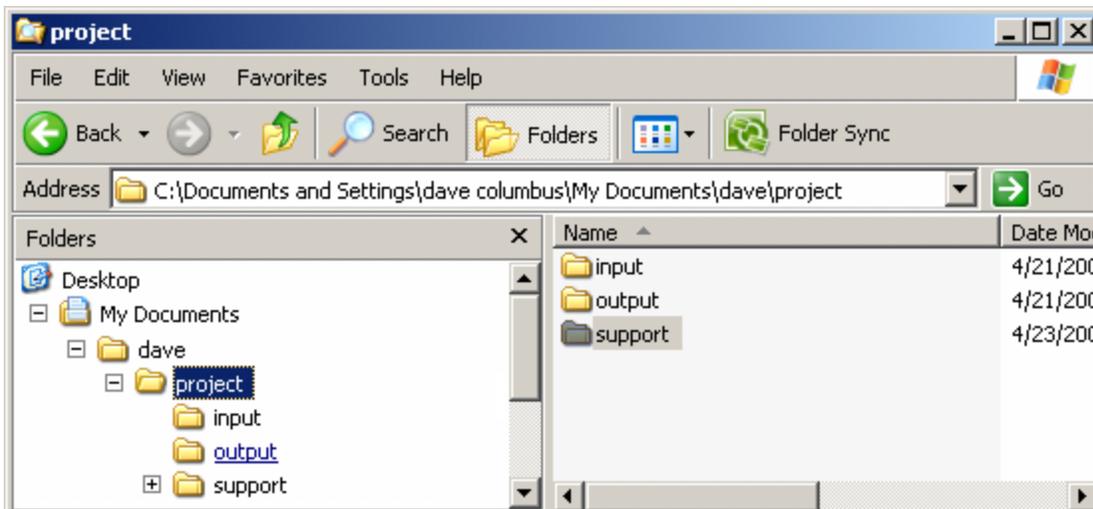
## Description

This is an overview of your first run of AutoMap3. It is not an all-inclusive view of AutoMap but will cover the, more or less, necessary basics.

## Procedure

### 1. Create Workspace

A good starting point is creating a project directory, a place where all your input files and output files will reside. This helps prevent files from getting lost. One suggestion is to create a top level project directory and create input and output directories within that directory. I also create a directory to place all support files such as Delete Lists and all the Thesauri.



Place all your input files into the **input** directory. Place any Delete Lists, Thesauri and other files to be used in the **support** directory. Direct all your results to the **output** directory.

## 2. Preparing Your Files

Files can come from any source. But to use in AutoMap3 they must be .txt files. Files in a **Word** format or **html** files will not be accepted into AutoMap. Any of these formats must be re-saved as a **.txt** format. This could be as simple as re-saving the file in the correct format to doing a copy-and-paste of the text into a text editor.

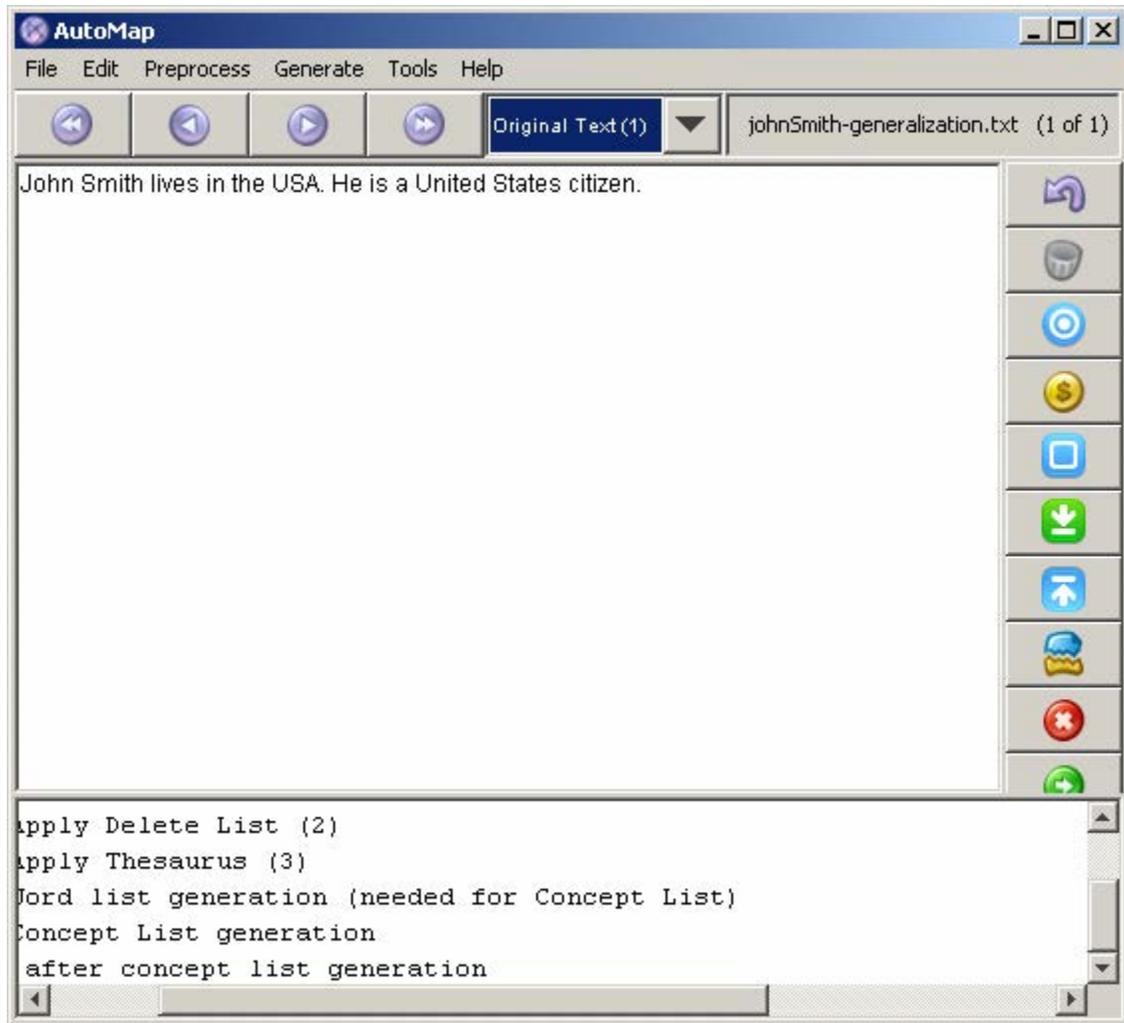
**NOTE :** *AutoMap accepts a variety of text encodings but they must be in a .txt format.*

## 3. Load your files into AutoMap

Once the text files are in the correct format they can be loaded into AutoMap. From the Pull Down Menu select **File = Select Input Directory** and navigate to the directory where you placed your text files. The first of these files will appear in the main window.

*usCitizen.txt*

John Smith lives in the USA. He is a United States citizen.



## 4. Decide on the Preprocessing Functions to use

Functions from the **Preprocessing Menu** affect all loaded text. None of these functions create any output files (though some require externally created files to work). These functions **remove excess concepts** (e.g. Remove Punctuation, Numbers, extra white spaces, or Symbols) or **modify concept names** (e.g. Thesauri for creating key\_concepts) for easier generation and post-processing functions.

## 5. Creating a Generalization Thesaurus

Many people, places and things can be known by a variety of different formats of their names. A General Thesauri helps consolidate these various names under one unifying term. Below is an example. The format is

**concept,key\_concept**. Concept can be one or more words but key\_concept must be one single word which can use the underscore.

#### *usCitizenGenThes.csv*

John Smith,John\_Smith  
United States,United\_States\_of\_America  
USA,United\_States\_of\_America

**NOTE** : The Thesaurus changes both the 2nd and 3rd concepts to the key\_concept **United\_States\_of\_America**.

Create this file in a text editor or spreadsheet program and save it as a **.csv** file.

## 6. Apply a Generalization Thesaurus

From the Pull Down Menu select **Preprocess => Apply Generalization Thesauri** and navigate to your Generalization Thesaurus. At the Adjacency dialog box select **Rhetorical**. After applying the text will change to reflect the application of the thesaurus.

#### *Text after General Thesauri applied*

John\_Smith lives in the United\_States\_of\_America. He is a United\_States\_of\_America citizen.

## 7. Edit your General Thesauri

After applying a Thesaurus the list can be altered by selecting from the Pull Down Menu **Tools => Thesauri Editor**.

## 8. Delete Extra/Unneeded Concepts

Texts usually have many extra concepts, or noise, that are not relevant to the semantic connections. A Delete List removes those extra words. From the Pull Down Menu select **Preprocess => Apply Delete List** and navigate to the Delete List you want to use. The text in the main will change to reflect the application of the Delete List.

#### *usCitizenDeleteList.txt*

in  
the  
is  
a

## *Text after Delete List with rhetorical adjacency*

John Smith lives ~~xxx~~ ~~xxx~~ USA. He ~~xxx~~ ~~xxx~~ United States citizen.

## 9. View and Alter Delete Lists

After creating Delete List you may want to make changes to it with the Delete List Editor. From the Pull Down Menu select **Tools => Delete List Editor**. From this tool you can add or remove concepts from a Delete List. From the Pull Down Menu select **File => Save as Delete List** and either replace the old file or save it as a new file.

**NOTE :** *If you decide to make changes to the Delete List then **Undo** the applied Delete List and reapply the new one.*

## 10. Generate Output

After preprocessing the text it's time to produce output from them. The Pull Down Menu **Generate** contains functions that tell AutoMap to write output for the function selected; Concept Lists, Semantic List, Parts of Speech, and other useful functions.

Each function outputs files that can be examined for analysis and used to further process the text files.

## 11. Example of Output for Concept List

From the Pull Down Menu select **Generate => Concept List**. AutoMap will ask for the directory to save the concept list.

```
concept,frequency,relative_frequency,gram_type,tf-idf
"He","1","0.5","single","0.0"
"John_Smith","1","0.5","single","0.0"
"United_States_of_America","2","1.0","single","0.0"
"citizen","1","0.5","single","0.0"
"lives","1","0.5","single","0.0"
```



## Description

This section contains general information about Files and Encodings



## Description

The File Menu contains functions whose main purpose is working with the files themselves. These functions do not perform any processing or generate any output. They work with raw files to help prepare them for use in AutoMap.

### Import Text Files

Allows you to load one or more files into AutoMap. You can either 1) select an entire directory or 2) select individual files to load. When selecting individual files you can select non-contiguous files by holding down the Control key while clicking the files to select. This is the only command which will actually bring text into AutoMap.

If your imported text is in the UTF-8 the first two lines in the Preprocessing Order Window will be identical. But when importing text in other formats.

### Original Text

□□S□y□n□o□p□s□i□s□:□ □T□h□e□  
 □T□o□k□'□r□a□ □p□l□a□n□ □t□o□  
 □k□i□l□l□ □a□l□l□ □t□h□e□ □S□y□  
 s□t□e□m□ □L□o□r□d□s□.□ □T□h□e□  
 □ □p□l□a□n□ □i□s□ □t□o□ □i□n□f□  
 l□i□t□r□a□t□e□ □t□h□e□ □s□u□m□  
 m□i□t□ □a□n□d□ □p□o□i□s□o□n□  
 □t□h□e□ □S□y□s□t□e□m□ □L□o□r□  
 d□s□.□ □B□u□t□ □t□h□e□y□ □n□  
 e□e□d□ □a□ □"□h□u□m□a□n□"□ □  
 w□h□o□ □s□p□e□a□k□s□ □g□o□u□  
 '□a□l□d□ □a□n□d□ □t□h□a□t□ □h□  
 u□m□a□n□ □i□s□ □D□a□n□i□e□l□  
 □J□a□c□k□s□o□n□ □o□f□ □t□h□e□  
 □S□G□C□.□ □H□e□ □s□p□e□a□k□s□  
 □ □g□o□u□'□a□l□d□.□ □T□h□e□ □T□  
 □o□k□'□r□a□ □a□p□p□r□o□a□c□h□  
 □ □D□a□n□i□e□l□,□ □t□h□e□ □S□G□

### Text Imported

Synopsis: The Tok'ra plan to kill all the System Lords. The plan is to infiltrate the summit and poison the System Lords. But they need a "human" who speaks gou'ald and that human is Daniel Jackson of the SGC. He speaks gou'ald. The Tok'ra approach Daniel, the SGC, and the U.S. Military, with their plan and he agrees. SG-1 and SG-17 travel with the Tok'ra to Revenna. After outlining the plan to Daniel, he is taken by Jacob Carter to the summit where he is posing as a low ranking Gou'ald. O'Neill stays on Revenna with SG-1 and SG-17. The assassination plan is proceeding fine until a new emissary, Osiris, appears. She recognizes Daniel but stays silent. Daniel and Jacob both know the assassination of the System Lords would now cause complications. Mean

**NOTE :** This function only works with text files.

**NOTE :** When using the script you still have to specify an entire directory.

AutoMap will present a dialog box asking for two parameters.

1. **Select Text Encoding :** This defaults to **Let AutoMap Detect** but you can change this to another encoding if you know the format of your files.
2. **Select Text Direction :** This defaults to **Left->Right, Top->Bottom** but you can change this if you know the direction of your text.

## Create New Text File

Creates a blank document in AutoMap which is useful in using copy-and-paste clips from multiple documents. After your new document is complete this file, with any other processing done to it, can be saved.

## File Conversion Functions

The functions in the [Conversion Section](#) help make files more compatible with AutoMap.

# File Save Functions

The functions in the [Save section](#) allow you to save various files from your work in AutoMap.

---

## Exit AutoMap

When you exit, AutoMap will ask if you want to save your preferences. Remember, there are two sets of preferences: **User and System**. This includes the directory you visited last, the options you used when you created a metanetwork (directionally, window, etc.), your font choices, and others. These will be restored the next time you start AutoMap.

**NOTE :** *It will not save the state of loaded files after exiting.*

After saving the preferences it will close all files and exit.



## Description

There are many types of text formats available. Only the text format with the .txt extension works correctly in AutoMap. If your data is in any other format it must be converted before using it in AutoMap.

## Thesauri Format

**Thesauri Format :**  
*conceptFrom,conceptTo*

# Master Thesauri Format

## **Master Thesauri Format :**

*conceptFrom,conceptTo,metaOntology,MetaName*

The Master Thesauri combines the generalization (**from and to**), the metanetwork (**to and meta**).

Marking for the Delete List is represented when the character [#] is placed in the meta column.

Any item which contains  as part of it's icon is related to a Master Thesauri funtion.

**metaOntology** is one of the ora types: **agent, organization, location, event, knowledge, resource, task.**

See [Content Overview => Ontology](#) for more information.

## **Example**

**conceptFrom** : United State of America

**conceptTo** : USA

**metaOntology** : location



## **Description**

A character encoding system consists of a code that pairs a sequence of characters from a given character set (sometimes incorrectly referred to as code page) with something else, such as a sequence of natural numbers, octets or electrical pulses, in order to facilitate the transmission of data (generally numbers and/or text) through telecommunication networks and/or storage of text in computers.

**UTF-8** : It is able to represent any character in the Unicode standard, yet is backwards compatible with ASCII. UTF-8 encodes each character (code point) in 1 to 4 octets (8-bit bytes), with the single octet encoding used only for the 128 US-ASCII characters. See the Description section below for details.

**NOTE** : *If empty boxes appear in the text this is an indication the text is using the Microsoft version of UTF-8 instead of the standard encoding.*

**Western** : A standard character encoding of the Latin alphabet. It is less formally referred to as Latin-1. It was originally developed by the ISO, but later jointly maintained by the ISO and the IEC. The standard, when supplemented with additional character assignments (in the C0 and C1 ranges: 0x00 to 0x1F and 0x7F, and 0x80 to 0x9F), is the basis of two widely-used character maps known as ISO-8859-1 (note the extra hyphen) and Windows-1252.

**UTF-16** : (Unicode Transformation Format) is a variable-length character encoding for Unicode, capable of encoding the entire Unicode repertoire. The encoding form maps each character to a sequence of 16-bit words. Characters are known as code points and the 16-bit words are known as code units. For characters in the Basic Multilingual Plane (BMP) the resulting encoding is a single 16-bit word. For characters in the other planes, the encoding will result in a pair of 16-bit words, together called a surrogate pair. All possible code points from U+0000 through U+10FFFF, except for the surrogate code points U+D800-U+DFFF (which are not characters), are uniquely mapped by UTF-16 regardless of the code point's current or future character assignment or use.

**GB2312** : The registered internet name for a key official character set of the People's Republic of China, used for simplified Chinese characters. GB abbreviates Guojia Biaozhun, which means national standard in Chinese.

**Big5** : The original Big5 character set is sorted first by usage frequency, second by stroke count, lastly by Kangxi radical. The original Big5 character set lacked many commonly used characters. To solve this problem, each vendor developed its own extension. The ETen extension became part of the current Big5 standard through popularity.

**NOTE** : *AutoMap uses the **Hard Return** to designate paragraph breaks.*

## Text Direction

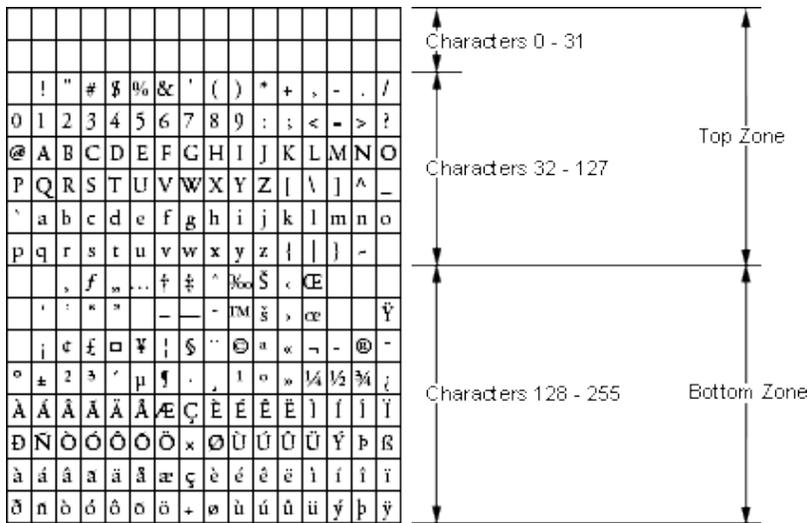
Languages can be written either left-to-right (LTR) or right-to-left (RTL). The majority of languages use a LTR syntax. The most notable RTL languages are Arabic and Hebrew.

## Single-Byte Fonts

Each character in a font uses a keystroke or combination of keystrokes to produce the character. Fonts based on most Western languages will have 256 possible characters. Each character in these fonts is one-byte in length. A one-byte character can have one of 256 possible values.

In a font, each character is arranged in a specific order. This is called a font's **encoding**, which is explained in more detail below. The computer uses this information to tell which character to display or print when a key is pressed. The spaces occupied by characters are called **slots**. Each slot represents a different keypress. For example, if you were working on a word-processing document and you were to hold the Shift key while pressing the letter "A" on the keyboard, you would see the letter **A** appear on the screen.

If you will notice, in the character map below, there are 2 **zones** - the top zone and the bottom zone. The top zone has characters 0 through 127 and the bottom zone has characters 128 through 255. The point to this is that characters occupying slots 32 through 127 in the top zone are identical in both Macintosh and Windows operating systems. Characters 0 through 31 (the "lower 32") are usually reserved for the operating system. The slots in the bottom zone (the **extended characters**) are different. These are the ones that will usually cause the problems.



**NOTE :** A font's encoding is simply a lookup table (an index) which is used to translate computer codes into the characters in the font.



## Encoding Lesson

## Encoding Problems

The first you need to know is when you use the **Select Input Directory** AutoMap expects to find files in the standard **UTF-8** format. If the files in the directory are encoded differently the text in the display will not show up correctly.

Because the text you want to analyze could possibly come from a variety of sources there's no assurance that it's in the UTF-8 format. Word files, web pages, emails, or whatever else you can find can have a variety of encodings. And sometimes when you import text you find it's **NOT** in the form you thought it would be.

## Those empty little boxes

Occasionally, when importing text, you will get empty boxes instead of the some specialty characters. This is due to differences in the encoding schemes. What is the problem? It's a simple explanation and a simple fix.

There are two kinds of UTF-8, with and without the BOM (Byte Order Mark) at the beginning of the file. Microsoft products require the BOM in order to recognize UTF-8, and the UTF-8 they produce has a BOM. Most other products produce UTF-8 without a BOM.

Technically no BOM is required for UTF-8, but Microsoft has adopted the convention of using its presence to distinguish that encoding from the OS default.

So without that marker some of your text may become mismarked.

## Cut-and-Paste problems

Your first document may have started out with a proper UTF-8 format. But as you begin cutting and pasting material from other sources you may be adding other formats. Anytime text is pasted into another document it retains its encoding.

## Smart Quotes

Smart quotes are not considered proper quote characters. The standard value for the straight quote character is **34**; there is only one character for both the beginning and ending quotes. The smart, or curly, quotes are actually two entirely different characters, **"** (left curly quote mark) and **"** (right curly quote mark).

The original designers of the ASCII character set did not define a standard method for identifying properly curved quotation marks, so computers have had a problem properly exchanging quotation marks ever since.

## The Solution

First try AutoMap's **Text Import** function and allow AutoMap to try to detect the source encoding. This function is designed to convert text into standard UTF-8 format.

If the program doesn't work, you can also use a third party text-editor to convert the data. Both Microsoft Word and the free Notepad++ text editor (<http://notepad-plus.sourceforge.net/uk/site.htm>) support this functionality. Simply open the file in your editor and do a **Save As...** a .txt file. Microsoft's Notepad saves files as single-byte ANSI (ASCII) by default.

# Foreign Characters Sets

Foreign characters sets are an entirely different matter. These will require you to have the proper font installed on your computer as each font can possibly be encoded differently.



## Description

The Table Viewer is used to view any **.csv** file. If the file contains headers they will be displayed at the top

## Generalization Thesauri - Standard Format

Antony	Marc_Antony
bad omens and portent	bad_omen
battlefield	Battlefield_Of_Philippi
battlefield of Philippi	Battlefield_Of_Philippi
Brutus	Marcus_Brutus
Brutus's	Marcus_Brutus

## Generalization Thesauri - Master Format

conceptFrom	conceptTo	metaOntology	metaName
Antony	Marc_Antony	agent	
bad omens and portent	bad_omen		
battlefield	Battlefield_of_Philippi	location	
battlefield of Philippi	Battlefield_of_Philippi	location	
Brutus	Marcus_Brutus	agent	

## Agent by Agent Square Matrix

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
A1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
A2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
A4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

# GUI

## Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

**NOTE :** *The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.*

## Pull-Down Menus

### File Menu



**Open File :** Navigate to a .csv file to view. If it's a compatible file the information will be displayed in the viewer.



**Save Message Log Window :** Saves the Message Log Window to the directory of your choice.

**Compare File :** After selecting your first table you can use this function compare another .csv file. This compare function works slightly different from other compare functions. Instead of examining an individual column to compare it does a one-to-one compare in list order. (e.g. item 1 of file 1 is compared to item 1 of file 2, and so on down the lists).



As in other compare functions a white background means the cell values are identical, a green background means the compare file is a new value, a red background means the compared cell doesn't exist in the loaded file, and a yellow background means the values are different.



**Exit :** Exits the Table Viewer and returns to the Main GUI.

## Edit Menu



**Compare File** : Compares a second Table to the currently loaded Table.



**Find** : Highlights in the table the searched for word.

**NOTE** : Searching for *t\*e* would find *the, there, and theatre* (if all three were in your list).



**Reset Colors** : Resets all colors to black text on white backgrounds.

**NOTE** : The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparison open a new file.



## Description

A viewer that can be used to view text files which have been tagged with either **Parts-of-Speech** or **MetaNetwork** tags.

## Parts of Speech Tagged File

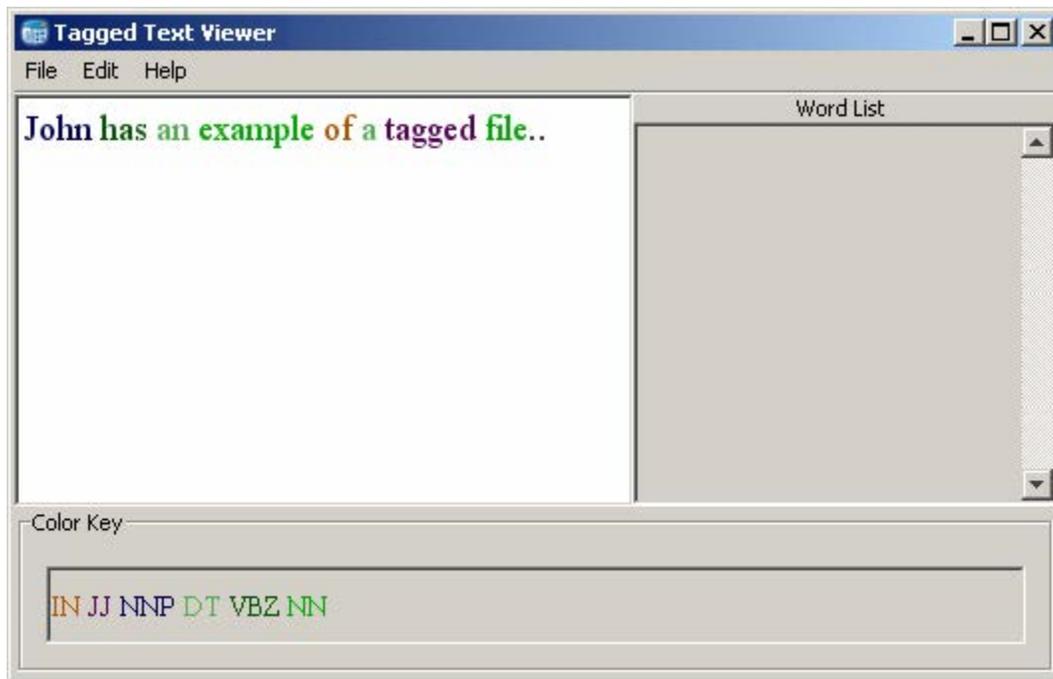
A Parts of Speech tagged file contains tags defining the part of speech of each concept. This is done from the main GUI **Generate => Parts of Speech Tagging**. The file created can be either in the **.txt or .csv** format. For use in the Tagged Text Viewer you need to save your file in the .txt format.

## aTaggedFile.txt

*John has an example of a tagged file.*

# POS Tags

John/NNP has/VBZ an/DT example/NN of/IN a/DT tagged/JJ file/NN  
./.



## MetaNetwork Tagged File

A MetaNetwork tagged file is generated from the main GUI menu **Generate => MetaNetwork => MetaNetwork Text Tagging**. First you will be asked to select a location to save your file. Then you will be asked to navigate to a MetaNetwork thesaurus to use.

## aTaggedFile.txt

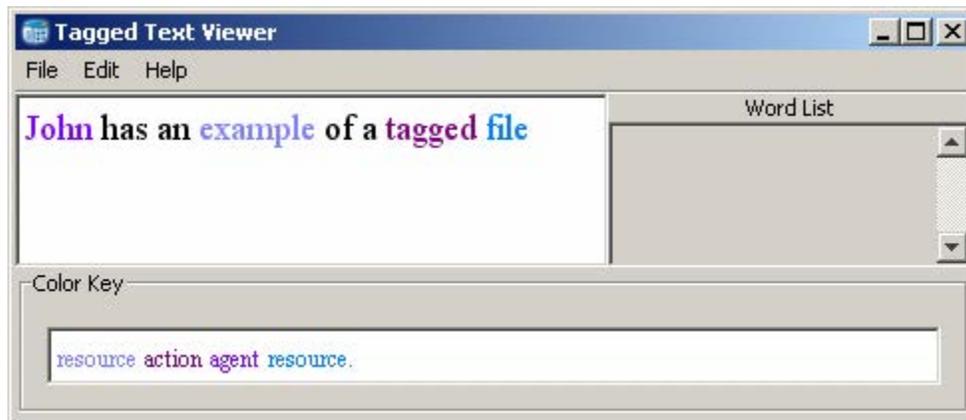
John has an example of a tagged file.

## MetaNetworkThesaurus.csv

John,agent  
example,resource  
tagged,action  
file,resource

# MetaNetwork Tags

John/agent has an example/resource of a tagged/action file/resource.



## GUI

### Word List

A list of words selected from the text is displayed in this pane. **Clicking** any of the words in the display window will place the word in the **Word List panel**.

### Color Key

The color coding of the concepts in the display window match the colors of the definitions in the Color Key at the bottom of the window. For a complete list of the Parts of Speech see [Content => Parts of Speech](#).

The Color Key can also be used to highlight various parts of speech in your text. By clicking on the parts of speech in the Color Key the corresponding tagged concepts will be highlighted in the text window.

### No Selections

Mike went to store and work  
V PRE NNP NN CON

## Select nouns (NN)

Mike went to store and work  
V PRE NNP NN CON

## Select proper nouns (NNP)

Mike went to store and work  
V PRE NNP NN CON

## Pull-Down Menus

### File Menu

-  **Open File** : Loads a text file into the viewer.
-  **Save as HTML** : Saves the current file in the HTML format. This file can be used for demonstration purposes or for purposes of further analysis.
-  **Save Checked Words to File** : Saves all checked words to a list.
-  **Add Checked Words to Delete List** : Places all checked words in a text file as a list with one word per line.
-  **Add Checked Words to MetaNetwork Thesaurus** : AutoMap asks you to classify the checked words and create a comma-delimited file. All the checked items will receive the same classification.
-  **Exit** : Exits the Tagged Text Viewer and returns to the Main GUI.

## Edit Menu



**Remove Checked Words** : Removes from the Word List all words that have been checked.



**Remove All Words** : Removes all words from the Word List regardless of whether or not they are checked.

---



**Identify Possible Misspellings** : Italicizes all words it deems as possible misspellings.



**Find Word** : Makes any instance of the found word bolds. This function can be repeated multiple times and previous found words will remain in bold. Use **Reset Colors** to clear.

**NOTE** : Searching for *t\*e* would find *the, there, and theatre* (if all three were in your list).

---



**Reduce Deleted Words** : Makes any instance of a deleted word reduced in size.

### Regular display

synopsis: xxx tok ra plan xxx kill xxx xxx system lords.  
xxx plan xxx xxx infiltrate xxx summit xxx poison xxx  
system lords

### Reduced display

synopsis: xxx tok ra plan xxx kill xxx xxx system lords. xxx plan  
xxx xxx infiltrate xxx summit xxx poison xxx system lords



**Show Delete List Impact** : Asks for a Delete List to apply and will display, by strike-through, how that Delete List would affect the file.



**Show Generalization Thesaurus Impact** : Asks for a Generalization Thesaurus and will display, by underlining adjacent

concepts, how that thesaurus would affect the file.



**Show MetaNetwork Thesaurus Impact** : Asks for a MetaNetwork Thesaurus and will display, by color coding found concepts.



**Set Font Size** : Changes the font size using HTML sizes, not point sizes.



**Set Font Style** : Allows you to change the display to any font on your computer.



**Reset** : Resets all colors and font styles in the display to their defaults.



## Description

This section contains general information about the use of Concept Lists



## Description

A Concept List is all the concepts of one individual file.

Using a Concept List a text can be refined using other functions such as a Delete List (to remove unnecessary concepts) and Generalization Thesaurus (to combine n-grams into single concepts).

## Example:

johnDoeInc.txt

John Doe works at John Doe Inc.

Concept List:

John, Doe, works, at, John, Doe, Inc

johnDoeIncDeleteList.txt

at

Concept List after Delete List applied. The concept **at** is now missing.

John, Doe, works, John, Doe, Inc

johnDoeIncGenThes.csv

John Doe Inc,John\_doe\_inc  
John Doe,john\_doe

After applying Generalization Thesaurus the concept list has fewer concepts but they are more meaningful. **John** and **Doe** are combined into the person's name **John\_Doe** as are the three individual concepts **John, Doe, & Inc.** into the name of the **John\_Doe\_Inc..**

john\_doe  
works  
john\_doe\_inc

**NOTE :** *The order of the concepts in the Generalization Thesaurus is important. See Order of thesauri entries under Thesauri, Generalization for more information.*

## Information obtained from a Concept List

### frequency

The number of times a concept was found in a file

### relative\_frequency

The frequency of any concept divided by the highest value obtained of any frequency.

### gram\_type

### tf-idf

term frequency–inverse document frequency - a statistical measure used to evaluate how important a word is to a document



## Description

The Union Concept List differs from the Concept List in that it considers concepts across all texts currently loaded, rather than only the currently selected text file. The Union Concept List is helpful in finding frequently occurring concepts, including those that, after review, can be added to a Delete List.

The Union Concept List includes:

- The concepts found in all files and the total frequency.
- Related, cumulative frequencies of concepts in all text sets.
- Cumulated unique concepts and total concepts contained in the data set.

**NOTE :** *The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.*

## Definitions

**Concept :** The individual concepts in the file.

**POS :** Defines the Parts of Speech of each concept

**Frequency :** Number of times a concept appears in a file.

**Relative Frequency :** The frequency of any concept divided by the highest value of any frequency

**Relative Percentage :** The result of adding all of the relative frequency values then dividing a concept's relative frequency by that value.

## Example

Start with two (or more) texts.

### ***johnIsAFireman.txt***

John is a Fireman in lower Manhattan in New York City. John was there at the Twin Towers on that day in September.

### ***nyc.txt***

NYC is a city comprised of five boroughs: Manhattan, Queens, the Bronx, Brooklyn, and Staten Island.

A Concept list for each input text:

#### ***fireman.csv***

City,1,0.33333334,single  
Fireman,1,0.33333334,single  
John,2,0.6666667,single  
Manhattan,1,0.33333334,single  
New,1,0.33333334,single  
September,1,0.33333334,single  
Towers,1,0.33333334,single  
Twin,1,0.33333334,single  
York,1,0.33333334,single  
a,1,0.33333334,single  
at,1,0.33333334,single  
day,1,0.33333334,single  
in,3,1.0,single  
is,1,0.33333334,single  
lower,1,0.33333334,single  
on,1,0.33333334,single  
that,1,0.33333334,single  
the,1,0.33333334,single  
there,1,0.33333334,single  
was,1,0.33333334,single

#### ***nyc.csv***

Bronx,1,1.0,single  
Brooklyn,1,1.0,single  
Island,1,1.0,single  
Manhattan,1,1.0,single  
NYC,1,1.0,single  
Queens,1,1.0,single  
Staten,1,1.0,single  
a,1,1.0,single  
and,1,1.0,single  
boroughs,1,1.0,single

city,1,1.0,single  
comprised,1,1.0,single  
five,1,1.0,single  
is,1,1.0,single  
of,1,1.0,single  
the,1,1.0,single

A Word list for each input file:

***fireman.csv***

John, is, a, Fireman, in, lower, Manhattan, in, New, York,  
City, John, was, there, at, the, Twin, Towers, on, that, day,  
in, September

***nyc.csv***

NYC, is, a, city, comprised, of, five, boroughs, Manhattan,  
Queens, the, Bronx, Brooklyn, and, Staten, Island

A unionConceptList.csv file using both files:

concept, frequency, relative\_frequency, relative\_percentage  
Bronx,1,0.5,0.125  
Brooklyn,1,0.5,0.125  
Island,1,0.5,0.125  
Manhattan,2,1.0,0.25  
NYC,1,0.5,0.125  
Queens,1,0.5,0.125  
Staten,1,0.5,0.125  
a,2,1.0,0.25  
and,1,0.5,0.125  
boroughs,1,0.5,0.125  
city,1,0.5,0.125  
comprised,1,0.5,0.125  
five,1,0.5,0.125  
is,2,1.0,0.25  
of,1,0.5,0.125  
the,2,1.0,0.25  
City,1,0.5,0.125  
Fireman,1,0.5,0.125  
John,2,1.0,0.25  
New,1,0.5,0.125  
September,1,0.5,0.125  
Towers,1,0.5,0.125  
Twin,1,0.5,0.125  
York,1,0.5,0.125

```
at,1,0.5,0.125
day,1,0.5,0.125
in,3,1.5,0.375
lower,1,0.5,0.125
on,1,0.5,0.125
that,1,0.5,0.125
there,1,0.5,0.125
was,1,0.5,0.125
```

This Union Concept List can be used as the basis for creating a Delete List or a MetaNetwork Thesauri (Carley, 2002) for all texts loaded.

## Using in Excel

A Union Concept List can be sorted in Excel. Open the file in Excel. All the data will appear in a single column. To separate it, first select the column with the data. Then select **Data => Text to Columns** from the menu. In the dialog box select **Delimited** and click Next. Select the check box for **Comma** and click Finish. The data is now in individual columns. To sort the list, highlight the data and select **Data => Sort....** Select "frequency" under "Sort by" and make sure it is descending. Then select concept under "Then by". Your Union Concept List is sort by frequency.



## Description

The **Concept List Viewer** is used to view and edit concept lists created from AutoMap. With the viewer you can sort the list by any of the headers. With the **Selected** column you can create a **Delete List**.

Select	concept	frequency	relative_frequency_across_texts	relative_percentage_across_texts	tf-idf	gram_type	number_of...
<input type="checkbox"/>	Calpurnia	6	0.05	0.0	0.00123	single	3.0
<input type="checkbox"/>	camp	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	can	2	0.02	0.0	7.4E-4	single	2.0
<input type="checkbox"/>	cannot	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	captive	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	captor	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	captured	3	0.02	0.0	0.00194	single	1.0
<input type="checkbox"/>	care	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	Casca	8	0.06	0.0	0.00295	single	2.0
<input type="checkbox"/>	Cassius	46	0.36	0.02	0.0	single	5.0
<input type="checkbox"/>	Cato	2	0.02	0.0	0.0013	single	1.0
<input type="checkbox"/>	cause	2	0.02	0.0	7.4E-4	single	2.0
<input type="checkbox"/>	celebrate	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	celebration	2	0.02	0.0	0.0013	single	1.0
<input type="checkbox"/>	certain	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	cheered	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	cheering	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	chooses	1	0.01	0.0	6.5E-4	single	1.0
<input type="checkbox"/>	Cicero	2	0.02	0.0	0.0013	single	1.0

\*\*\* MESSAGE WINDOW \*\*\*

## Columns

**Select** : Selected items are the ones AutoMap performs any processing functions on.

**concept** : Each individual concept is contained on a separate row.

**frequency** : The amount of occurrences found for that concept.

**relative\_frequency\_across\_text** :

**relative\_percentage\_across\_text** :

**tf-idf** :

**gram\_type :**

**number\_of\_texts :** The number of texts in which a particular concept is found.

## GUI

### Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

**NOTE :** *The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.*

## Pull-Down Menus

### The File Menu

 **Open File :** Select a Concept List to load into the Viewer.

See [Compare Concept Lists](#) lesson for more information.

 **Save Message Log :** Saves the message log in the Concept List window.

 **Save as Delete List :** Saves check items as a new Delete List.

 **Exit :** Exits the Concept List Viewer and returns to the Main GUI.

### The Edit Menu

 **Compare File :** Compares a second Concept List to the currently loaded Concept List.

-  **Properties** : Display the **Total Concepts** and the **Unique Concepts** in the loaded file.
-  **Select All** : Places a check mark in every box in the **Select** column.
-  **Select None** : Removes the check marks from every box in the **Select** column.
-  **Select Minimum Threshold** : Selects all concepts with frequencies equal to or greater than the Minimum Threshold.
-  **Select Maximum Threshold** : Selects all concepts with frequencies equal to or less than the Maximum Threshold.
-  **Find** : AutoMap asks for term to locate. If there are any matches the background of the found item will be colored blue.

**NOTE** : Searching for *t\*e* would find **the, there, and theatre** (if all three were in your list).

-  **Reset Colors** : To end the comparison use **Reset** and all the color bands will be removed.

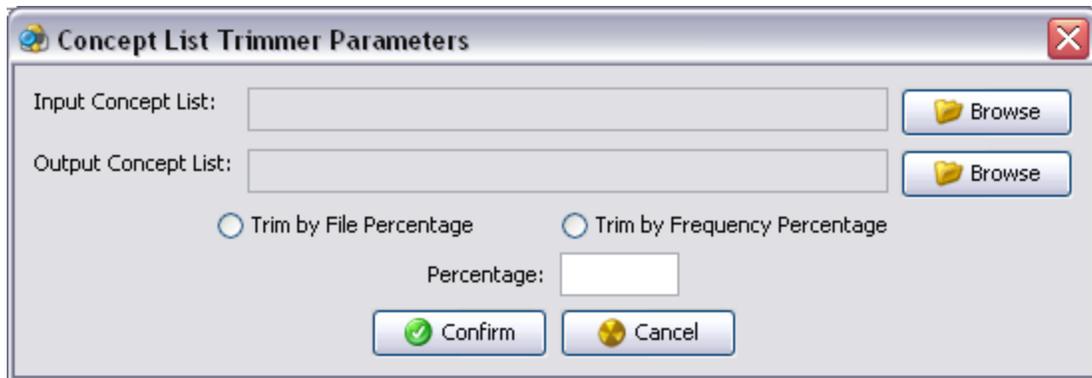
**NOTE** : The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparison open a new file.

## Procedures

**Concept List Trimmer** : Removes a percentage of the list by one of two methods.

**Trim by File Percentage** : Trims the concept list by removing lowest frequency items based on percentage of file. Enter 10 and the lowest 10% will be removed.

**Trim by Frequency Percentage** : Trims the concept list by removing lowest frequency items by based on their frequency as a percentage of highest frequency item. Enter 10 and if the highest frequency item is 100 then every concept that occurs 10 or fewer times will be removed.



## Using a Concept List

### Description

Concept Lists (the frequency of concepts in one file) and Union Concept Lists (the frequency of concepts throughout all loaded files) can be used to work with concepts from one or more text files. It lists the frequency, relative frequency, and gram type. When working with multiple files you'll find the Union Concept List useful.

**NOTE :** *The number of unique concepts considers each concept only once. The number of total concepts considers repetitions of concepts.*

### Concept List Procedure

#### 1. Select a text file(s) to use

Place your text file(s) in an empty directory. Load the file(s) by selecting from the Pull Down Menu **File => Import Text Files**.

*theBoy-4.txt:*

```
See the boy named Dave. He has 2 balls. 1 ball is red. 1 ball  
is blue.
```

*milkAndCookies.txt:*

```
Dave wants milk and cookies. He drives to the store. He then
buys milk and cookies.
```

## 2. Create Concept List

From the Pull Down Menu select **Generate => Concept List => Concept List (per text)**. Navigate to where you want to save the file(s) and click **Select**.

## 3. Decide if you need Union Concept List

After specifying the directory for the Concept List(s) AutoMap will ask if you want to create a Union Concept List. Unless you know you will not need it, click **Yes**.

## 4. Review a Concept List in the Concept List Viewer

The Concept List(s) and Union Concept List can be viewed using **Concept List Viewer**. From the Pull Down Menu select **Tools => Concept List Viewer**. From the Viewer Pull Down Menu select **File => Open File** and navigate to the location of the concept list to view.

Select	concept ▲	frequency	relative_frequency	gram_type	tf-idf
<input type="checkbox"/>	and	2	1.0	single	0.4620981203732...
<input type="checkbox"/>	buys	1	0.5	single	0.2310490601866...
<input type="checkbox"/>	cookies	2	1.0	single	0.4620981203732...
<input type="checkbox"/>	Dave	1	0.5	single	0.0
<input type="checkbox"/>	drives	1	0.5	single	0.2310490601866...
<input type="checkbox"/>	he	1	0.5	single	0.2310490601866...
<input type="checkbox"/>	He	1	0.5	single	0.0
<input type="checkbox"/>	milk	2	1.0	single	0.4620981203732...
<input type="checkbox"/>	store	1	0.5	single	0.2310490601866...
<input type="checkbox"/>	the	1	0.5	single	0.0
<input type="checkbox"/>	Then	1	0.5	single	0.2310490601866...
<input type="checkbox"/>	to	1	0.5	single	0.2310490601866...
<input type="checkbox"/>	wants	1	0.5	single	0.2310490601866...

\*\*\* MESSAGE WINDOW \*\*\*

**NOTE :** If you load a Union Concept List the right-most column has the header **relative\_percentage** denoting the frequency of a concept occurring in all text files.

## Concept List Viewer functions

### Sorting a Concept List

A Concept List can be sorted by clicking on any of the headers. This will sort the list in an ascending order. Clicking the same header again will reverse the sorting order to descending. Which header is being used will be denoted by a small triangle to the left of the header name.

**NOTE :** The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.

### Creating a Delete List

A Delete List can be created from the Concept List Viewer. Place a check mark in the **Selected** column of each concept to include in a Delete List. From the Pull Down Menu select **File => Save as Delete List**. AutoMap will prompt you to select a location to save the Delete List. Give it a unique filename and click **Open**.

**NOTE :** *The Delete List can be saved in either the .txt or a .csv format.*

This new Delete File can now be loaded and applied to your texts.

## Selecting Concepts

In the Viewer **Edit** menu the viewer gives you options for selecting/unselecting multiple concepts.

**Select All :** Places a check mark in every check box.

**Select None :** Clears all check marks from the concept list.

**Select Minimum Threshold :** The number input will select concepts with frequencies equal to or greater than the threshold to be selected.

**Select Maximum Threshold :** The number input will select concepts with frequencies equal to or lesser than the threshold to be selected.

**Find :** will highlight the input text, if found, with a cyan background.

**Reset Colors :** removes all highlighting.



## Description

Compares two Concept Lists and displays concepts that appear in either both files (white background) or in a single file (red or green background). The viewer is called from the pull-down menu by selecting **Tools => Concept List Viewer**.

## Load

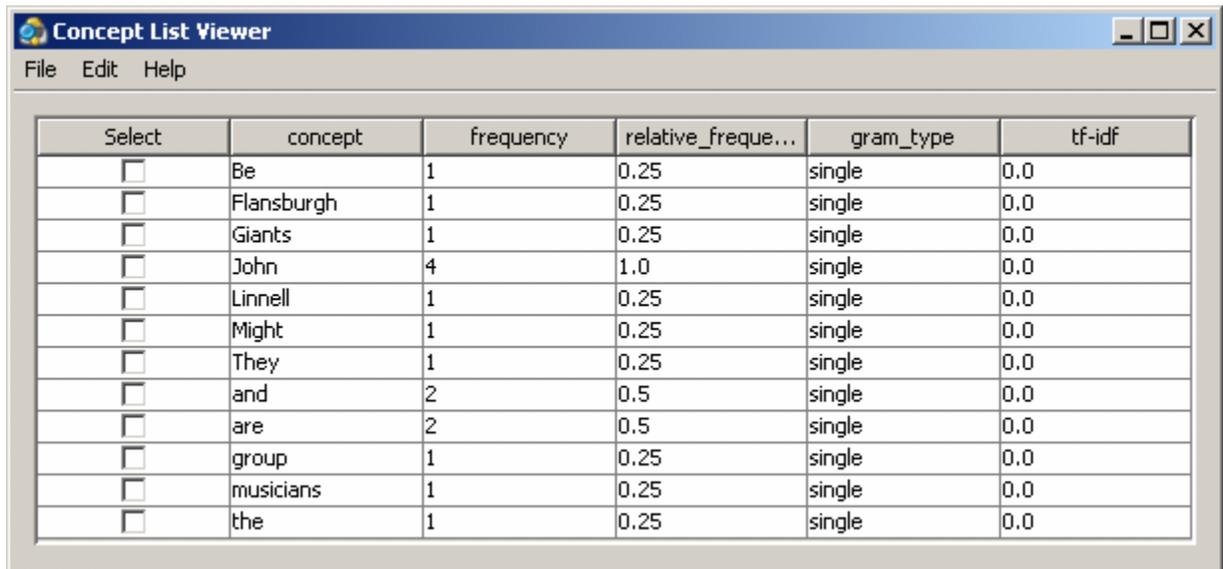
This example will use a Concept List created with a raw file and a Concept List created after applying a thesaurus. This will be used to demonstrate how the viewer detects frequency differences.

## tmbg.txt

```
John Flansburgh and John Linnell are musicians. John and John  
are the group They Might Be Giants.
```

The tmbg's file is loaded into AutoMap with **File => Import Text Files** and a Concept List is created **Generate => Concept List => Concept List (Per Text)**. This Concept List is then loaded into AutoMap with **Tools => Concept List Viewer**.

**NOTE :** Notice the concept **John** has a frequency of **4**.



The screenshot shows a window titled "Concept List Viewer" with a menu bar containing "File", "Edit", and "Help". Below the menu bar is a table with the following columns: "Select", "concept", "frequency", "relative\_frequ...", "gram\_type", and "tf-idf". The table contains 14 rows of data, each with a checkbox in the "Select" column.

Select	concept	frequency	relative_frequ...	gram_type	tf-idf
<input type="checkbox"/>	Be	1	0.25	single	0.0
<input type="checkbox"/>	Flansburgh	1	0.25	single	0.0
<input type="checkbox"/>	Giants	1	0.25	single	0.0
<input type="checkbox"/>	John	4	1.0	single	0.0
<input type="checkbox"/>	Linnell	1	0.25	single	0.0
<input type="checkbox"/>	Might	1	0.25	single	0.0
<input type="checkbox"/>	They	1	0.25	single	0.0
<input type="checkbox"/>	and	2	0.5	single	0.0
<input type="checkbox"/>	are	2	0.5	single	0.0
<input type="checkbox"/>	group	1	0.25	single	0.0
<input type="checkbox"/>	musicians	1	0.25	single	0.0
<input type="checkbox"/>	the	1	0.25	single	0.0

Close the Concept List Viewer.

Go to the directory where your Concept List is saved and rename it tmbg-noThes.csv. This will prevent AutoMap from overwriting the file.

Apply the generalization thesaurus tmbg-genThes.csv file with **Preprocess => Apply Generalization Thesauri**.

## tmbg-genThes.csv

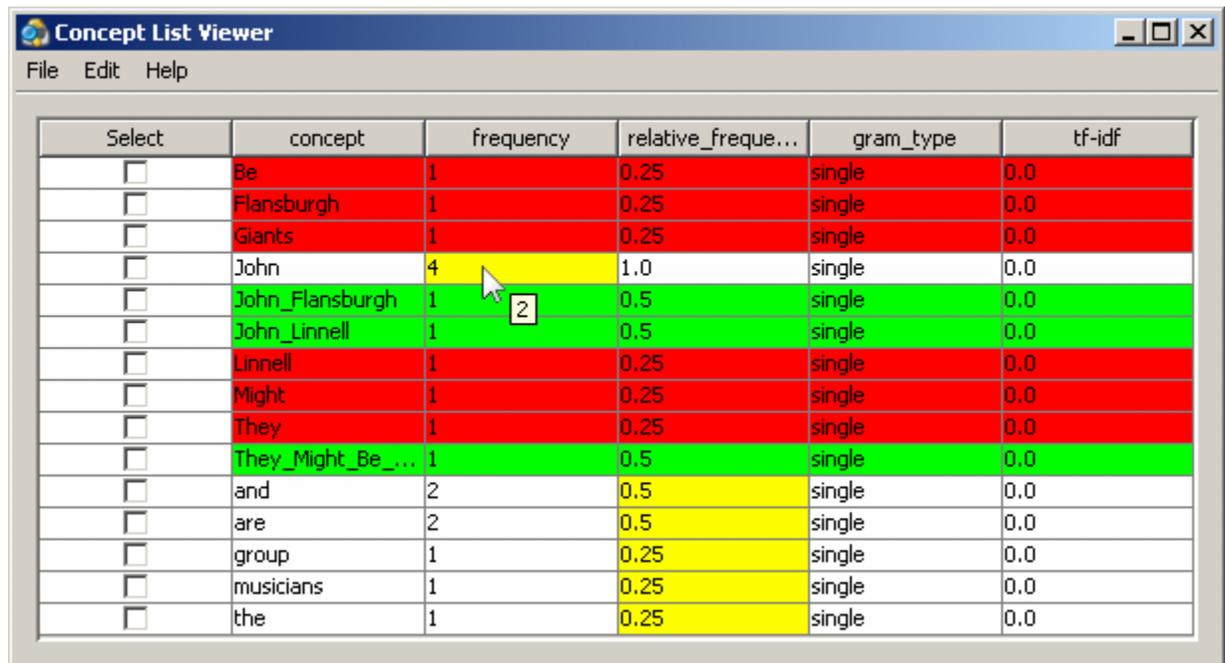
```
John Flansburgh,John_Flansburgh  
John Linnell,John_Linnell
```

They Might Be Giants,They\_Might\_Be\_Giants

Create a Concept List with **Generate => Concept List => Concept List (Per Text)**. This Concept List, named tmbg.csv, will be used as the compare file.

## Comparing the two files

Start the Concept List Viewer with **Tools => Concept List Viewer**. Load the first Concept List with **File => Open File** and navigate to the directory containing the tmbg-noThes.csv (the file without the thesaurus applied). Next we will compare this with the second file. From the pull-down menu select **Edit => Compare File** and navigate to the directory containing tmbg.csv (the file with the thesaurus applied).



Select	concept	frequency	relative_frequ...	gram_type	tf-idf
<input type="checkbox"/>	Be	1	0.25	single	0.0
<input type="checkbox"/>	Flansburgh	1	0.25	single	0.0
<input type="checkbox"/>	Giants	1	0.25	single	0.0
<input type="checkbox"/>	John	4	1.0	single	0.0
<input type="checkbox"/>	John_Flansburgh	1	0.5	single	0.0
<input type="checkbox"/>	John_Linnell	1	0.5	single	0.0
<input type="checkbox"/>	Linnell	1	0.25	single	0.0
<input type="checkbox"/>	Might	1	0.25	single	0.0
<input type="checkbox"/>	They	1	0.25	single	0.0
<input type="checkbox"/>	They_Might_Be_...	1	0.5	single	0.0
<input type="checkbox"/>	and	2	0.5	single	0.0
<input type="checkbox"/>	are	2	0.5	single	0.0
<input type="checkbox"/>	group	1	0.25	single	0.0
<input type="checkbox"/>	musicians	1	0.25	single	0.0
<input type="checkbox"/>	the	1	0.25	single	0.0

The concepts with white backgrounds are found in both files. The concepts with red backgrounds are found only in the original file. The concepts in green are found only in the newly compared file.

The cells with the yellow backgrounds are concepts found in both files (e.g. concepts on white backgrounds) but they have different values. Notice on the image below the arrow hovering over the frequency value for **John**. In the original file John had a value of **4**. The tooltip displays a value of **2** which is the value in the compared file.

# Message Log Window

After making a comparison AutoMap will display the statistics of the comparison in the Message Log Window. It will display added (green), deleted (red), and altered (yellow).



## Description

This section contains general information about using Thesauri.



## Description

The Generalization Thesauri are used to replace possibly confusing concepts with a more standard form (e.g. a text contains United States, USA and U.S. The Generalization Thesauri could have three entries which replace all the original entries with united\_states). Creating a good thesaurus requires significant knowledge of the content.

## Format of a Thesauri

1. Every line contains a concept found in the text followed by the concept to replace it with. The syntax is **some old concept,some\_old\_concept**
2. The **original** concept can be one or more words in a row.

3. A **Key** concept **must** be one word.
4. The **original** concept and the **key** concept are separated with a comma.
5. There should not be any space before or after the comma.
6. The Thesaurus is not case sensitive.

## Uses for a Generalization Thesauri

### Combining multi-word concepts

Peoples names usually consist of two or more individual names like John Smith or Jane Doe.

`John Smith becomes John_Smith.`

It is also useful if, after the initial presentation of the full name, a person is referred to by only part of that name. The thesauri would be able to create one concept out of either entry.

`John Smith becomes John_Smith`

`John becomes John_Smith.`

### Normalizing abbreviations

Many large companies and organizations are recognized by the abbreviation of their name as well as the name itself.

`The British Broadcasting Company is routinely known as the BBC.`

`The Chief Executive Officer of a company is known as the CEO.`

**NOTE** : Be aware that some ordinary words can be misinterpreted as organizations. One notable example is **WHO - World Health Organization**.

## Normalizing contraction

Contractions are used to shorten two concepts into one smaller concept.

`isn't => is not | I'd => I would | they'll => they will`

Expanding these contractions out to their roots allows for creating better Delete Lists.

## Correcting typos

When typing people routinely make small spelling errors. Many of these are done when people are not sure of the correct spelling.

`absense,absence | centruy,century | manuever,maneuver`

Or correcting common typing mistakes

`hte instead of the | chaor instead of chair`

## Globalizing countries

For some countries there are multiple ways to refer to it's name. America, for example, has many ways to reference it's name.

`US | U.S. | United States | United States of America | America`

`Germany | Deutschland (German) | Allemagne (French) | Niemcy (Polish)`

Creating a thesauri entry for each of these will reduce the number of concepts in a file while grouping all the same concepts, with variate names, in the same frequency.

Each set can be contained in a separate thesauri and run on a set of texts individually.

## Example:

### *johnInUSA.txt*

My name is John Smith and I live in the USA.

### *johnInUSAGenThes.csv*

John Smith,John\_Smith  
USA,United\_States

### *Text after GenThes applied:*

My name is John\_Smith and I live in the United\_States.

### *Thesauri Content Only*

Thesauri Content Only creates an output using ONLY the entries found in the thesauri. All other concepts are discarded.

**NOTE :** When using this option you need to be aware of what is, and is not, in the thesauri.

## **Example with ThesauriContentOnly not activated**

### *johnInUSA.txt*

My name is John Smith and I live in the USA.

### *johnInUSAGenThes.csv*

John Smith,John\_Smith  
USA,United\_States

### *Text after Generalization Thesauri applied:*

My name is John\_Smith and I live in the United\_States.

## **Example using ThesauriContentOnly**

### *TextjohnInUSA.txt*

My name is John Smith and I live in the USA.

### *johnInUSAGenThes.csv*

John Smith,John\_Smith  
USA,United\_States

## *Text after Generalization Thesauri applied with ThesauriContentOnly:*

John\_Smith United\_States.



## Description

Meta-Network (Carley, 2002) associates text-level concepts with Meta-Network categories {agent, resource, knowledge, location, event, group, task, organization, role, action, attributes, when}. One concept might need to be translated into several Meta-Network categories. For example, the concept commander corresponds with the categories agent and knowledge.

The top level of the meta-network ontology is who, what, how, where, why, when. All concepts can be fit to one of these categories.

## Meta-Network categories

**agent** : A person, group, organization, or artificial actor that has information processing capabilities. All "who"s are agents, be they a person in a group, a group within an organization, or the organization itself (e.g. President Barack Obama, the shadowy figure seen outside the building, or the Census bureau). Which sub-category the agents are placed in is left to the user.

**knowledge** : Information learned such as a school lecture or knowledge learned from experience (e.g. Excellent knowledge of the periodic table or "I know what you did last summer").

**resource** : Can be either a physical or intangible object. A resource is anything that can be used for the completion of a job. (e.g. One uses a car to drive from point A to point B and money to fund a terrorist organization).

**task** : A task is part of a set of actions which accomplish a job, problem or assignment. Task is a synonym for activity, although the latter carries a connotation of being possibly longer duration

**event** : Something that happens, especially something of importance. Events are usually thought of as a public occasions, but they can also be clandestine meetings. The number of agents can range in the thousands or as few as two agents (e.g. Christmas in Times Square or dinner with friends).

**organization** : A group of agents working together for a common cause (e.g. The Red Cross or the local chess club).

**location** : An actual physical place. This could be a room in a building, a city, or a country (e.g. Pittsburgh, PA or my living room).

**role** : An agent's role can be defined as their job for their employer or the part they serve during an event.

**action** : driving to the mall, eating lunch. Used as a verb.

**attribute** : Information about the specifics of the agents. These are usually traits that agents have in common, each can be slightly different (e.g. visible traits like hair colour or intangible traits like religious beliefs).

**when** : Referring to time or circumstances. Can be as broad as a year or as pinpoint as the exact time of a particular day (e.g. Last year or 2:33 PM on March 1st, 2009).

## Example:

Let's take two short sentences as an example. It contains **people, places, and things**

*dairyFarm.txt*

```
Ted runs a dairy farm. He milks the cows, runs the office, and  
cleans the barn.
```

### *dairyFarmDeleteList.txt*

There are some unnecessary concepts in the text. Using a Delete List will extract the essence of the text. This Delete List is quite short.

```
a, and, in, on, the
```

After applying the delete list, the text appears in the display like this:

```
Ted runs xxx xxx dairy farm. He milks xxx cows, runs xxx  
office, xxx cleans xxx barn.
```

### *Meta-Network Thesaurus:*

Now we come to the meta-network thesaurus. This file will define the category for each of the **important** concepts we have in the file.

### *dairyFarmMeta.csv*

```
Ted,agent  
runs,task  
dairy,resource  
farm,location  
He,agent  
milks,task  
cows,resource  
office,location  
cleans,task  
barn,location
```

### *Examining the File*

Generating a DyNetML file in AutoMap prepares it to be examined in ORA.



## Description

**Thesaurus Content Only** is an option used with the **Generalization Thesaurus**. It allows you to select how your results will be display and output.

## **synopsis-2.txt**

Synopsis: The Tok'ra plan to kill all the System Lords. The plan is to infiltrate the summit and poison the System Lords. But they need a "human" who speaks gou'ald and that human is Daniel Jackson of the SGC. He speaks gou'ald. The Tok'ra approach Daniel, the SGC, and the U.S. Military, with their plan and he agrees. SG-1 and SG-17 travel with the Tok'ra to Revenna. After outlining the plan to Daniel, he is taken by Jacob Carter to the summit where he is posing as a low ranking gou'ald. O'Neill stays on Revenna with SG-1 and SG-17. The assassination plan is proceeding fine until a new emissary, the gou'ald Osiris, appears. She recognizes Daniel but stays silent. Daniel and Jacob both know the assassination of the System Lords would now cause complications. Meanwhile Revenna is attacked. O'Neill, Carter, Teal'c, and Elliot help in the defense of the planet. Daniel escapes the summit. He joins up with Jacob and they make their escape back to Revenna intending to rescue O'Neill and SG-1. Their craft is shot down. Elliot sacrifices his life in order to allow SG-1 to escape.

## **synopsis-2GenThes.csv**

assassination plan,assassination\_plan  
Carter,Maj\_Samantha\_Carter  
Daniel,Daniel\_Jackson  
Daniel Jackson,Dr\_Daniel\_Jackson  
Elliot,Lt\_Elliot  
gou'ald,gou\_ald  
Jacob,Jacob\_Carter  
Jacob Carter,Jacob\_Carter  
low ranking gou'ald,low\_ranking\_gou\_ald  
O'Neill,Col\_Jack\_O'Neill  
SG-1,SG1  
SG-17,SG17  
speaks gou'ald,speak\_gou\_ald  
summit meeting,summit  
System Lord,System\_Lords  
System Lords,System\_Lords  
Teal'c,Teal\_c  
the SGC,Stargate\_Command  
Tok'ra,Tok\_ra  
U.S. Military,US\_Military

**Thesaurus Content Only - NO** : Selecting **NO** will retain all concepts. Thesaurus concepts will be replaced and the entire text will be displayed in the window. Below is an example with the replaced thesaurus entries in bold.

Synopsis: The **Tok\_ra** plan to kill all the **System\_Lords**. The plan is to infiltrate the summit and poison the **System\_Lords**. But they need a "human" who **speak\_gou\_ald** and that human is **Dr\_Daniel\_Jackson** of **Stargate\_Command**. He **speak\_gou\_ald**. The **Tok\_ra** approach **Daniel\_Jackson**, **Stargate\_Command**, and the **US\_Military**, with their plan and he agrees. **SG1** and **SG17** travel with the **Tok\_ra** to Revenna. After outlining the plan to **Daniel\_Jackson**, he is taken by **Jacob\_Carter** to the summit where he is posing as a **low\_ranking\_gou\_ald**. **Col\_Jack\_O\_Neill** stays on Revenna with **SG1** and **SG17**. The **assassination\_plan** is proceeding fine until a new emissary, the **gou\_ald** Osiris, appears. She recognizes **Daniel\_Jackson** but stays silent. **Daniel\_Jackson** and **Jacob\_Carter** both know the assassination of the **System\_Lords** would now cause complications. Meanwhile Revenna is attacked. **Col\_Jack\_O\_Neill**, **Maj\_Samantha\_Carter**, **Teal\_c**, and **Lt\_Elliot** help in the defense of the planet. **Daniel\_Jackson** escapes the summit. He joins up with **Jacob\_Carter** and they make their escape back to Revenna intending to rescue **Col\_Jack\_O\_Neill** and **SG1**. Their craft is shot down. **Lt\_Elliot** sacrifices his life in order to allow **SG1** to escape.

**Thesaurus Content Only - YES** : Selecting **YES** will eliminate all concepts that do not exist in the thesaurus. The results will depend on a second option chosen.

## Thesaurus content only options:

**Direct adjacency** : All non-thesaurus concepts will be removed from the display and be replaced with a space.

: Tok\_ra System\_Lords. summit System\_Lords. " speak\_gou\_ald  
Dr\_Daniel\_Jackson Stargate\_Command. speak\_gou\_ald. Tok\_ra  
Daniel\_Jackson, Stargate\_Command, US\_Military, . SG1 SG17  
Tok\_ra . Daniel\_Jackson, Jacob\_Carter summit  
low\_ranking\_gou\_ald. Col\_Jack\_O\_Neill SG1 SG17.  
assassination\_plan , gou\_ald , . Daniel\_Jackson .  
Daniel\_Jackson Jacob\_Carter System\_Lords . . Col\_Jack\_O\_Neill,  
Maj\_Samantha\_Carter, Teal\_c, Lt\_Elliot . Daniel\_Jackson  
summit. Jacob\_Carter Col\_Jack\_O\_Neill SG1. . Lt\_Elliot SG1 .

**Rhetorical adjacency** : Non-thesaurus concepts are removed from the display but are replaced with a **(xxx)** placeholder. This will show the distance between the thesaurus items.

```
xxx: xxx Tok_ra xxx xxx xxx xxx System_Lords. xxx xxx xxx
xxx xxx xxx summit xxx xxx xxx System_Lords. xxx xxx xxx xxx
"xxx" xxx speak_gou_ald xxx xxx xxx xxx Dr_Daniel_Jackson xxx
Stargate_Command. xxx speak_gou_ald. xxx Tok_ra xxx
Daniel_Jackson, Stargate_Command, xxx xxx US_Military, xxx xxx
xxx xxx xxx xxx. SG1 xxx SG17 xxx xxx xxx Tok_ra xxx xxx. xxx
xxx xxx xxx xxx Daniel_Jackson, xxx xxx xxx xxx Jacob_Carter
xxx xxx summit xxx xxx xxx xxx xxx xxx low_ranking_gou_ald.
Col_Jack_O'Neill xxx xxx xxx xxx SG1 xxx SG17. xxx
assassination_plan xxx xxx xxx xxx xxx xxx xxx, xxx gou_ald
xxx, xxx. xxx xxx Daniel_Jackson xxx xxx xxx. Daniel_Jackson
xxx Jacob_Carter xxx xxx xxx xxx xxx xxx System_Lords xxx xxx
xxx xxx. xxx xxx xxx xxx. Col_Jack_O'Neill,
Maj_Samantha_Carter, Teal_c, xxx Lt_Elliot xxx xxx xxx xxx xxx
xxx xxx. Daniel_Jackson xxx xxx summit. xxx xxx xxx xxx
Jacob_Carter xxx xxx
Col_Jack_O'Neill xxx SG1. xxx xxx xxx xxx xxx. Lt_Elliot xxx
xxx xxx xxx xxx xxx xxx SG1 xxx xxx.
```



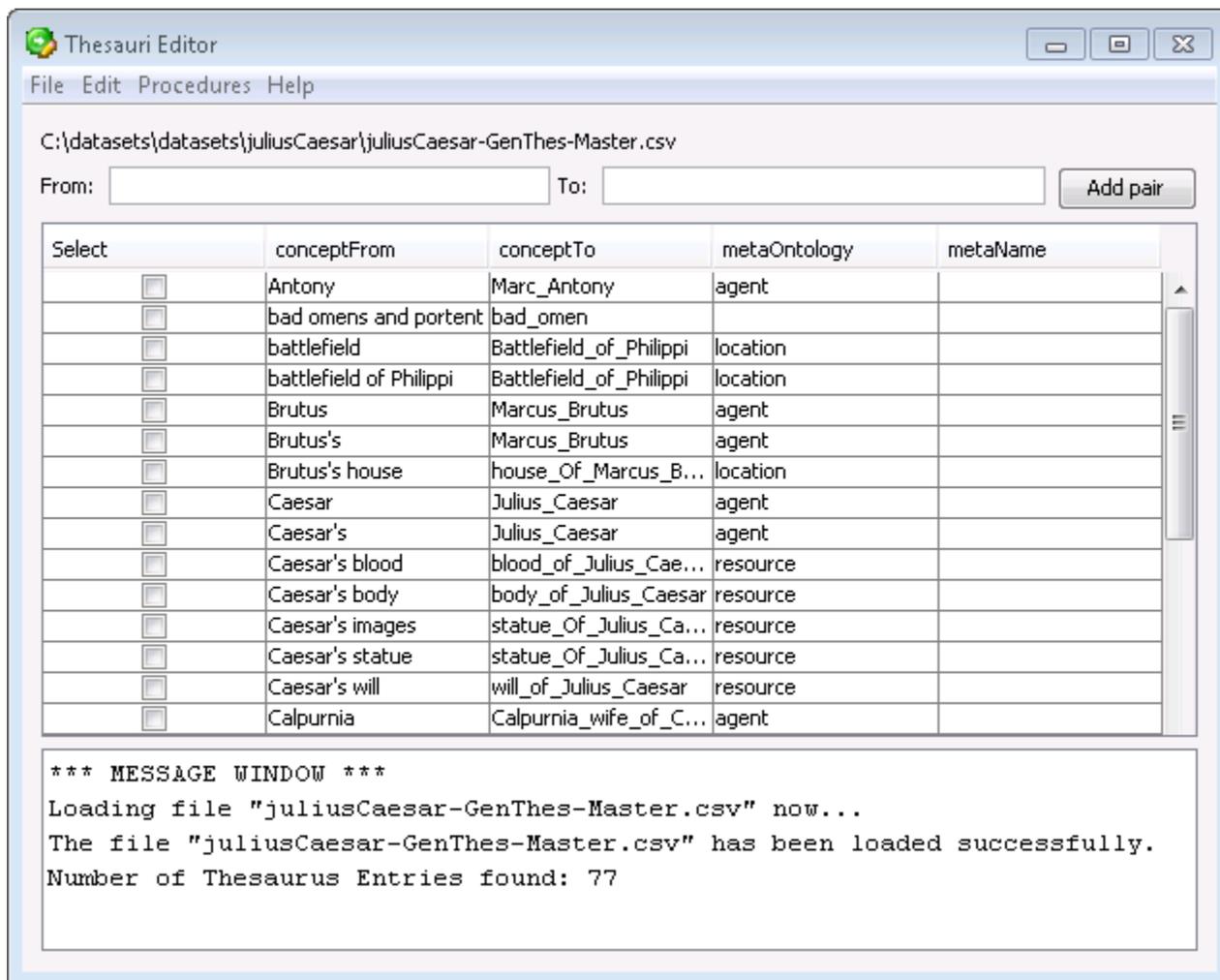
## Description

The Thesauri Editor can load and modify existing thesaurus files. Pairs of concepts can be added or subtracted. It can be compared to another thesaurus. Finally it can be saved under a new name.

Under the menu is displayed the name of the currently loaded file. It contains the full pathway of the file.

Below that are the **From:** and **To:** textboxes with the **[Add Pair]** button. This these tools you can add rows to the current file.

The main display contains five columns. **Select** is used to tell AutoMap which files to run Edit and Procedures on. **conceptFrom** contains the text as it appears in the original file. **conceptTo** is the concept you want to change it to. **metaOntology** contains the class of node. See [Content > Ontology](#) for more information. **metaName** for future use.



## GUI

If you find a pair that does not exist in your thesaurus it can be added by placing the raw text in the To: textbox and the key\_concept in the From: textbox. Then click the **Add pair** button to add it to the list.

## Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header

again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

**NOTE** : *The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.*

## The File Menu

 **Open File** : Select a Thesaurus to load into Editor.

See [Compare Thesauri Files Lesson](#) for more information

 **Save as...** : Saves the Thesaurus.

 **Save as...** : Saves a Thesaurus with a new name and/or to a new directory.

 **Save message Log** : Saves message log form the Thesaurus window.

 **Convert File to UTF-8** : Attempts to convert an input file into the UTF-8 format.

 **Exit** : Exits the Thesauri Editor and returns to the Main GUI.

## The Edit Menu

 **Compare** : Compares a second Thesaurus to the currently loaded Thesaurus.

 **Add Terms from Concept List** : Asks user to select a Concept List which will be added to the currently loaded Thesaurus.

 **Add Terms from NGram** : Asks user to select an NGram List which will be added to the currently loaded Thesaurus.

 **Add Stemmed Terms** : Adds stemmed words to the currently open

Thesaurus. The User will be asked whether to use the Porter Stemmer or the K-Stemmer.



**Select All** : Places a check mark in every box in the **Select** column.



**Select None** : Removes the check marks from every box in the **Select** column.



**Remove Selected** : Removes the concepts which contained a check mark in the Select column. The original file remains unaffected.



**Identify Possible Misspelling** : Highlights in yellow concepts AutoMap may consider misspelled. Hovering over these concepts will give a list of alternatives.



**Find** : AutoMap asks for term to locate. If there are any matches the background of the found item will be colored blue.

**NOTE** : *In a large thesaurus manually looking through it is usually not an option. Use the **Find** option and type in your search parameters in the textbox. The found item will be displayed with a blue background.*

**NOTE** : *Searching for **t\*e** would find **the, there, and theatre** (if all three were in your list.*



**Reset Colors** : To end the comparison use **Reset** and all the color bands will be removed.

**NOTE** : *The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparison open a new file.*

## The Procedures Menu

The functions in this pull-down menu do not affect the currently loaded Delete List. They are identical to the functions that can be found in the Main GUI.



**Apply Stemming to Thesauri** : You are asked to select a stemmer to apply (Porter Stemmer or K-Stem). All newly stemmed words will be added to the Thesaurus on screen. You need to use one of the **Save options** to keep this new list.,/tr>

-  **Merge Generalization Thesauri** : Allows you to select two or more Thesauri and combine them into one. AutoMap will then prompt you to save the new Thesaurus with a new name and location.
-  **Sort Thesaurus** : Choose a thesaurus to sort. AutoMap sorts the thesaurus by **number of words** (e.g. the more words in a concept then higher in the list it rises).
-  **Check Thesaurus for Missing Entries** : Verifies that each entry in a thesaurus contains no blanks before or after the comma. The line(s) containing the errors will be displayed in the message pane.
-  **Check Thesaurus for Duplicate Entries** : Will give the user a notice if there are duplicate entries in a thesaurus.
-  **Check Thesaurus for Circular Logic**: Will find each instance of Circular Login in a thesaurus and report the line(2) with the problems. Then it will report the total number of instances found.
-  **Check Thesaurus for Conflicting Entries**:

	<h2 style="margin: 0;">Compare Thesauri Files</h2>
---	--

## Description

The **Compare File** functions is implemented in many of the tools: **Delete List Editor, Thesauri Editor, Concept List Viewer, and Table Viewer**. It will compare the currently loaded file with a second file on disk and display if an entry is contained in both files (white background), originally loaded file (red background), or the compared file (green background).

**NOTE** : *The example uses thesauri files but the color coding is identical across all the included tools.*

## The Files

Below are the files used in this lesson. They can be copied and saved and used for your examples. Be sure to save them in the .csv format.

abc.csv	xyz.csv	a.csv	az.csv	a-caps.csv
a,alpha b,beta c,charlie	x,x-ray y-yankee z,zulu	a,alpha	a,alpha z,zulu	a,ALPHA

## The Comparisons

### Open a file

Start the Thesaurus Editor from the pull-down menus *Tools => Thesauri Editor*. From the Editor pull-down menu open the file **abc.csv**.

Select	To	From
<input type="checkbox"/>	a	alpha
<input type="checkbox"/>	b	beta
<input type="checkbox"/>	c	charlie

### Comparing a sub-set file

From the editor pull-down menu select *File => Compare Files*. Select the file **a.csv**. This file contains a single concept from the original **abc.csv** file.

Select	To	From
<input type="checkbox"/>	a	alpha
<input type="checkbox"/>	b	beta
<input type="checkbox"/>	c	charlie

The line **a,alpha** in white shows this concept is found in both files. The next two lines **b,beta; c,charlie** show these concepts are found in the original file but not in the compared file.

### Compare to an exclusive file

From the pull-down menu select *File => Compare Files* and select the file **xyz.csv**. The **xyz.csv** file contains no concepts that are found in the **abc.csv** file.

Select	To	From
<input type="checkbox"/>	a	alpha
<input type="checkbox"/>	b	beta
<input type="checkbox"/>	c	charlie
<input type="checkbox"/>	x	x-ray
<input type="checkbox"/>	y	yankee
<input type="checkbox"/>	z	zulu

The green lines for **x,x-ray; y,yankee; z,zulu** show these concepts are only found in the new file. The red lines for **a,alpha; b,beta; c,charlie** show these concepts were found in the original file but are absent from the compared file.

## A Super-set

From the pull-down menu select **File => Compare Files** and select the file **abcxyz.csv**. The compared file is the super-set of the original file which contains all the original concepts plus some unique concepts.

Select	To	From
<input type="checkbox"/>	a	alpha
<input type="checkbox"/>	b	beta
<input type="checkbox"/>	c	charlie
<input type="checkbox"/>	x	x-ray
<input type="checkbox"/>	y	yankee
<input type="checkbox"/>	z	zulu

The white lines **a,alpha; b,beta; c,charlie** show these concepts are contained in both files. The green lines **x,x-ray, y,yankee; z,zulu** show these concepts are found only in the compared file.

## Different values

From the pull-down menu select **File => Compare Files** and select the file **a-caps.csv**. The compared file contains only one similar term to the original file but the key\_concept is different.

There will be times when the same concept has different values in each file. This is defined by a yellow background. The value will display what is in the original file. By hovering over the cell it will display in a tooltip the value contained in the compared file.

Select	To	From
<input type="checkbox"/>	a	alpha
<input type="checkbox"/>	b	beta
<input type="checkbox"/>	c	charlie

## A Mixed Thesaurus

The most common comparison will be a mixture of terms that appear, or are missing from, each file.

From the pull-down menu select **File => Compare Files** and select the file **az.csv**. This file contains all the concepts from the **abc.csv** plus three additional concepts.

Select	To	From
<input type="checkbox"/>	a	alpha
<input type="checkbox"/>	b	beta
<input type="checkbox"/>	c	charlie
<input type="checkbox"/>	z	zulu

The white line for **a,alpha** shows this concept was found in both the original and compared files. The red lines **b,beta; c,charlie** show these concepts were found in the original file but not the compared file. The green line **z,zulu** shows this concept exists only in the compared file.



## Description

Thesauri are generally used to take multiple concepts, in different forms, and compile them under one **key concept**. If this is not done then the same concept could be listed many times as individual concepts.

## Thesauri Procedure

### 1. Select your text file(s)

Copy these files into a text editor and save them as **johnSmith.txt** and **countryThesauri.csv**. Place the johnSmith.txt in a folder by itself. Place

countryThesauri.csv in an accessible folder other than where johnSmith.txt resides.

### *usCitizen.txt*

```
John Smith lives in the USA. He is a United States citizen.
```

## 2. Prepare your Thesauri

For this simple file the thesauri is short. Larger texts could easily have hundreds of entries. This can be done in either a text editor or a spreadsheet.

### *usCitizenGenThes-Ext.csv*

```
The United States of America,United_States
United states,United_States
John Smith,John_Smith
America,United_States
USA,United_States
```

## 3. Loading the files

From the main menu select **File => Import Text Files**. Navigate to the directory where you placed johnSmith.txt and click **Select**. The file will appear in the main window.

## 4. Applying the thesauri

From the main menu select **Preprocess => Apply Generalization Thesauri**. Navigate to the directory where you placed countryThesauri.csv and click **Select**. The thesauri will be applied and a new text will appear in the window with the thesauri substitutions. **Apply Thesauri (3)** will now appear in the dropdown menu denoting AutoMap has applied the thesauri.

### *Text after thesauri applied*

```
John_Smith lives in the United_States. He is a United_States
citizen.
```

## Thesauri Editor

AutoMap contains a **Thesauri Editor** making it easy to revise your thesauri files. From the dropdown menu select **Tools => Thesauri Editor**.

From the Thesauri Editor menu select **File => Open File**. Navigate to the directory with your thesauri and click **[Okay]**. From the Thesauri Editor you can make changes to the thesauri.

## Questions regarding Thesauri

### Different Thesauri for different purposes

You might initially think it's necessary to keep all the thesauri entries in one place. Just easier to find everything. But in essence it would make it easier to keep track by splitting up the thesauri into smaller files, each with its own purpose. Below are some of the specialty thesauri used in this lesson.

You might have a general countries thesauri that is always used to fix multiple ways of posting a country. The U.S. can be listed in a multitude of ways:

`U.S., America, United States, The United States of America`

A country thesauri could have all these permutations listed and convert them all to the same concept say, `United_States`. So instead of four individual concepts you would have one.

`U.S.,United_States`  
`America,United_States`  
`United states,United_States`  
`The United States of America,United_States`

If your project concentrates on a particular field you could have a thesauri that contains names of organizations, resources, or people's names which appear with regularity.

Analysts working on a similar subject every day would need the same names, places, and resource for each run. A special thesauri could take care of that and would be easier to maintain as a single file.

### Is the thesaurus case-specific?

**NO!** If there are two or more entries for the same concept the first thesaurus entry will be used for all replacements.

`HE, Tom`  
`He, Dick`

he ,Harry

Every instance of **HE, He, and he** will be replaced with **Tom** without comparing their case.

## Running the Delete List Before or After the Thesauri

You may also find it necessary to run a Delete List on your files. Whether you apply a delete list before or after applying a generalization thesauri will depend on your set of files. A longer discussion can be found in the Content section under **Process Sequencing**.

## Using of one large thesauri vs. multiple smaller thesauri

This will be a personal choice of the user. Multiple thesauri have the advantage of easier editing but the downside is needing to apply multiple thesauri to the same set of files.

**NOTE :** *When doing multiple runs on the same data some analysts prefer to maintain a single thesaurus.*

**NOTE :** *The Generalization Thesaurus is NOT case sensitive to what it finds in the text. United States, United states, and united States are all considered the same bi-gram and would be replaced with the same entry.*



### Description

When working with a large amount of documents, each one being large themselves, you will find your thesauri growing increasing large. There are several items which you should be aware of in this circumstance.

## The Order of the Thesaurus

The order of the thesaurus entries can have an impact on your results. Better results will be obtained if the thesaurus is in a descending length order.

The **Sort Thesaurus** function can assist in ordering your thesauri. From the pull-down menu select *Procedures => Sort Thesaurus*. Navigate to the thesaurus you want to sort. Then give it a new name, and if you want, a new location and save it.

The Sort Thesaurus functions sorts by number of words in an entry. The order of entries with the same number of words will not change.

[See The AutoMap GUI > Procedures Menu for more information](#)

## Conflicts in Data

A problem can occur when you have more than one agent or location with the same name. In the beginning the text may specifically refer to **John Smith** and **John Doe** but afterwards you might find that a person is only referred to as **John**. AutoMap will use the first occurrence of John it finds when making substitutions.

AutoMap processes a thesaurus from top to bottom. This is why it's important to sort a thesaurus by the length of the entries.

## Sequence of Operations

`Mohammed abd al-Sha'bai`

This name contains both punctuation and symbols. Depending on the order of the preprocessing you could end up with any of the following three:

`Mohammed_abd_al_Sha_bai`

`Mohammed_abd_al-Sha'bai`

`Mohammed_abd_alShabai`

## How General Is Too General?

If you start with a 2,000 page text set, you will probably be unable to read all of the text prior to starting the processing. One of the problems you may encounter in such situations relates to common names and terms. For example, a preliminary review of the text may reveal that the documents contain an individual named **Joe**. Given that this is very common name, the data may contain several other individuals named **Joe**. Should this occur the program will incorrectly process all Joes as if they were the first Joe. Users need to exhibit caution before including common names and common terms in their thesauri.

If your text contains two names, *Joe Smith* and *Joe Jones* and a thesaurus entry for the name *Joe* then AutoMap will substitute all Joes for the first entry in the thesaurus.

## Using the **ThesauriContentOnly** option

You create a Meta-Network (Carley, 200) with the one-grams **dog, cow, and farm**. If you are going to use the **UseThesauriContentOnly** option then those three terms need to be in your General Thesaurus also. If they are not in the thesaurus then they will be eliminated from the output and the Meta-Network will not see them to tag them.

## Positive or Negative Links

AutoMap creates links between nodes. But AutoMap does not differentiate between positive and negative links. After processing that is up to the analyst.

*The U.S. lacks formal diplomatic relations with North Korea, Bhutan, and Cuba, but has close relationships with Canada and the U.K.*

Using a large window will create a fully connected graph with all links the same type. It will also make connections for all the countries involved. The reality is that there should be positive links to Canada and the U.K. and negative links to North Korea, Bhutan, and Cuba.

Using a small window size would create a link set that is wrong:

*US => North Korea => Bhutan => Cuba => Canada => UK*

*right:*

negative links: US => North Korea; US => Bhutan; US => Cuba

positive links: US => Canada; US => UK



## Description

This section contains general information about using Semantic Lists.



## Description

Semantic Lists contain pairs of concepts and their frequency in the chosen text file(s).

## Direction

**Uni-directional** : Will only look forward in the text file for a relationship. Any concept that came before will be ignored.

**Bi-Directional** : Will attempt to find a relationship in either direction of the concept. Both are constrained by windowSize and textUnit.

```
agent1 xxx agent2 xxx agent3.
```

Using **uni-directional and a window size of 3** agent2 would have a relationship to agent3 but not agent1. Relationships can only look forward in the text.

Using **bi-directional and a window size of 3** agent2 would have a relationship to both agent3 and agent1

**NOTE :** Using *bidirectional* can substantially increase the size of the *Semantic List*. A file with 17 concepts and using a window of 2 produced a *unidirectional Semantic List* of 13 entries whereas the *bidirectional Semantic List* consisted of 26 entries.

## Window Size

The distant concepts can be and still have a relationship to one another. Only concepts in same window can form statements. The window is defined in **textUnit**.

## Text Unit

The text unit can be comprised of one of the following:

**Sentence :** a sentence is a grammatical unit of one or more words.

**Word :** A word is a unit of language that represents a concept which can be expressively communicated with meaning

**Clause :** A clause consists of a subject and a verb. There are two types of clauses: independent and subordinate (dependent).

An **independent clause** consists of a subject verb and also demonstrates a complete thought: for example, "I am sad".

A **subordinate clause** consists of a subject and a verb, but demonstrates an incomplete thought: for example, "Because I had to move".

**Paragraph :** A paragraph is indicated by the start of a new line. It consists of a unifying main point, thought, or idea accompanied by supporting details.

**All :** The entire text



## Other Preprocessing

### Description

This section contains general information about Preprocessing.



## Preprocessing Menu

### Description

Following is a short description of the preprocessing functions in AutoMap3. These functions serve to prepare files to deliver output by reducing unneeded and unwanted concepts.

More detailed information can be found in the Content section as well as the individual tutorials and lessons.

### Undo Last Step

Undo removes the **last** Preprocessing done to the text. This is done one step at a time. To remove multiple preprocessing steps you must perform multiple undos.

### Redo All Steps

Reprocesses all Preprocessing steps. Useful if new text files are added or a support file has been altered.

## **Preform All Cleaning**

Contains functions for basic text clean up. Remove Extra Spaces, Fix Common Typos, Convert British to American Spelling, Expand Common Contractions, Expand Common Abbreviations, and Replace HTML Symbols.

## [Text Cleaning Functions Descriptions](#)

## **Preform All Preparation**

Contains functions to further prepare text. Pronoun Resolution, Remove American Letters, NGram Conversion, Remove Pronouns, Remove Noise Verbs, Remove Presitions, and Remove All Noise Words.

## [Text Preparation Functions Descriptions](#)

### **Text Refinement**

Contains the functions to finalize text. Remove Num bers, Remove Punctuation, Remove User Symbol, Remove Single Symbol, Remove Symbols, Convert to Lowercase, Convert to Uppercase, Apply Stemming, Apply Delete List, Apply Generatlization Thesauri.

## [Text Refinement Functions Descriptions](#)



### **Description**

An anaphoric expression is one represented by some kind of deictic, a process whereby words or expressions rely absolutely on context. Sometimes this context needs to be identified. These definitions need to be specified by the user. Used primarily for finding personal pronouns, determining who it refers to, and replacing the pronoun with the name.

**NOTE :** *Not all anaphora are pronouns and not all pronouns are anaphora.*

## Definition of Anaphora

Repetition of the same word or phrase at the start of successive clauses.

### milkAndCookies.txt

Dave wants milk and cookies. **He** drives to the store. **He** then buys milk and cookies.

The **He** at the beginning of the last two sentences are anaphoric under the strict definition (he refers to Dave).

## What is NOT an anaphora

Not all pronouns are anaphoras. If there is no reference to a particular person then it remains just a pronoun.

**He** who hesitates is lost.

The **He** at the beginning is NOT an anaphora as it does not refer to anyone in particular.



## Description

BiGrams are two adjacent concepts in the same sentence. Two concepts are not considered a bigram if they are in separate sentences or paragraphs. If a Delete List is run previous to detecting bi-grams then the concepts in the Delete List are ignored. Multiple Delete Lists can be used with a set of files.

## Definitions

### Frequency:

the number of times that bi-gram occurs in a single text.

### Relative Frequency:

The number of times a bi-gram occurs in a single text divided by the maximum occurrence of any bi-gram.

### Maximum Occurrence:

The number of times that the bi-gram that occurred the most, occurred in a text.

### Relative Percentage:

The percentage of all bi-grams accounted for by the occurrence of this bi-gram.

## The Most Common BiGram

Not all bigrams are important. In fact, the most common bigram, **of the**, is usually very unimportant by itself.

For example, in the movie title **Lord of the Rings** the important words are **Lord** and **Rings**. But without the bigram **of the** the title would make no sense: **Lord Rings**. By itself **of the** has no meaning, but within another set of words helps create the proper context.

## Changes in Meaning

When individual concepts are formed into bigrams their meanings can change.

## Threshold in regards to BiGrams

Threshold is used to detect if there are specific number of occurrences of a Bi-Gram in the text(s). For **Global Threshold** a Bi-gram is detected if the total number of its occurrences in all texts is greater than or equal to the Global Threshold. For **Local Threshold** a Bi-gram is detected if the number of its occurrences in EACH text is greater than or equal to the Local Threshold.

## Thresholds Example

GlobalThreshold=5 and LocalThreshold=2

```
text1: bi-gram X occurs 2 times  
text2: bi-gram X occurs 3 times
```

text3: bi-gram X occurs 1 time

The bigram "x" qualifies for GlobalThreshold:  $2+3+1 \geq 5$  (GlobalThreshold), but it doesn't qualify for LocalThreshold, because for text3 it occurs  $1 < 2$  (LocalThreshold) times.

## Bi-gram list

Here is an example.

### *fireman.txt*

```
John is a fireman.
```

#### ***Bi-Grams:***

```
John,is  
is,a  
a,fireman
```

## Bi-grams List using Delete List and Generalization Thesaurus

This is an example of how a Delete List and Generalization Thesaurus can affect the final bi-gram list.

### *associations.txt*

```
John Doe is actively involved in several industry and civic  
associations.
```

### *associationsDeleteList.txt*

```
is, in, and
```

### *associationsThesaurus.csv*

```
John Doe,John_Doe  
industry,business  
civic associations,business
```

### ***Using just the Delete List:***

```
John Doe actively involved several industry civic associations
```

***The bi-grams list:***

John,Doe  
Doe,actively  
actively,involved  
involved,several  
several,industry  
industry,civic  
civic,associations

***Using just the Generalization Thesaurus:***

John\_Doe is actively involved in several business and business

***The bi-grams list:***

John\_Doe,is  
is,actively  
actively,involved  
involved,in  
in,several  
several,business  
business,and  
and,business

***Using both the Delete List and the Generalization Thesaurus:***

John\_Doe actively involved several business business

***The bi-grams list:***

john\_Doe,actively  
actively,involved  
involved,several  
several,business  
business,business

## **Bi-Gram Chart**

The sample text and following chart show the relationship of frequency and relative frequency of the concepts in the text.

***businessLeader.txt***

John Doe is a business leader. John Doe is a president of the  
John Doe business.

Delete the noise from the text.

## *businessLeaderDeleteList.txt*

is a of the

Both **John** and **Doe** have a frequency of **3**. The bigram **John Doe** also have a frequency of **3**. This shows these concepts are important as both individual words and the bigram they create.

Words	Frequency	Relative Frequency	Relative Percentage
John	3	1	.3
Doe	3	1	.3
Business	2	.67	.2
Leader	1	.33	.1
President	1	.33	.1
Total Words	10		
Bi-Grams	Frequency	Relative Frequency	Relative Percentage
John Doe	3	1	.37
Doe business	2	.67	.25
business leader	1	.33	.12
Doe president	1	.33	.12
president John	1	.33	.12
Total bi-grams	8		8



## Description

Format Case changes the output text to either all lower or upper case.

## Example

[Sentence case](#)

Only the first word of the sentence and proper nouns are capitalized.  
My name is John Smith and I live in the USA.

#### Lower case

All letters are lowercase, even proper nouns.  
my name is john smith and i live in the usa.

#### Upper case

All letters are uppercase, even proper nouns.  
MY NAME IS JOHN SMITH AND I LIVE IN THE USA.

#### Title case

The first letter of every word is capitalized.  
My Name Is John Smith And I Live In The USA.

**NOTE :** *The problem with converting text is it disables the ability of Parts of Speech to correctly identify certain parts - such as Proper Nouns.*



## Description

Named-Entity Recognition allows you to retrieve proper names numerals, and abbreviations from texts.

## Items it Detects:

- Single words that are capitalized (e.g. Copenhagen).
- Adjacent words that are capitalized (e.g. The New York City Police Department).
- A string of adjacent words that are capitalized, but can be intervened by one non-capitalized word. The first and the last word in this string are capitalized (e.g. Canadian Department of National Defense).



## Description

Parts of Speech assigns a single best **Part of Speech**, such as noun, verb, or preposition, to every word in a text.

While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words match multiple tags, depending on the context that they appear in.

***Example :** Wind, for example, can be a noun in the context of weather, and can be a verb that refers to coiling something.) DeRose (DeRose, 1988) reports that over 40% of the words are syntactically ambiguous.*

Parts of Speech is often necessary before other functions are performed specifically when creating a Meta-Network (Carley, 2002). This Parts of Speech tagger is based on the **Hidden Markov Model**.

## The Hidden Markov Model

A Hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters; the challenge is to determine the hidden parameters from the observable data. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. An HMM can be considered as the simplest dynamic Bayesian network.

## Penn Tree Bank (PTB) Parts of Speech Table

<b>CC</b>	Coordinating conjunction	<b>PRP\$</b>	Possessive pronoun
<b>CD</b>	Cardinal number	<b>RB</b>	Adverb
<b>DT</b>	Determiner	<b>RBR</b>	Adverb, comparative
<b>EX</b>	Existential there	<b>RBS</b>	Adverb, superlative
<b>FW</b>	Foreign word	<b>RP</b>	Particle
<b>IN</b>	Preposition or subordinating	<b>SYM</b>	Symbol

	conjunction		
<b>JJ</b>	Adjective	<b>TO</b>	to
<b>JJR</b>	Adjective, comparative	<b>UH</b>	Interjection
<b>JJS</b>	Adjective, superlative	<b>VB</b>	Verb, base form
<b>LS</b>	List item marker	<b>VBD</b>	Verb, past tense
<b>MD</b>	Modal	<b>VBG</b>	Verb, gerund or present participle
<b>NN</b>	Noun, singular or mass	<b>VBN</b>	Verb, past participle
<b>NNS</b>	Noun, plural	<b>VBP</b>	Verb, non-3rd person singular present
<b>NNP</b>	Proper noun, singular	<b>VBZ</b>	Verb, 3rd person singular present
<b>NNPS</b>	Proper noun, plural	<b>WDT</b>	Wh-determiner
<b>PDT</b>	Predeterminer	<b>WP</b>	Wh-pronoun
<b>POS</b>	Possessive ending	<b>WP\$</b>	Possessive wh-pronoun
<b>PRP</b>	Personal pronoun	<b>WRB</b>	Wh-adverb

## Aggregate Parts of Speech

The PTB divides verbs into six subgroups (base form verbs, present participle or gerund verbs, present tense not 3rd person singular verbs, present tense 3rd person singular verbs, past participle verbs, past tense verbs). In some applications you might want to aggregate these into one verb group. Also, for certain purposes, the union of all prepositions, conjunctions, determiners, possessive pronouns, particles, adverbs, and

interjections could be collected into one group that represents irrelevant terms.

## Aggregation of PTB Categories

Aggregated Tag	Meaning	Number of Categories in PTB	Instances in PTB
<b>IRR</b>	Irrelevant term	16	409,103
<b>NOUN</b>	Noun	2	217,309
<b>VERB</b>	Verb	6	166,259
<b>ADJ</b>	Adjective	3	81,243
<b>AGENTLOC</b>	Agent	1	62,020
<b>ANA</b>	Anaphora	1	47,303
<b>SYM</b>	Noise	8	36,232
<b>NUM</b>	Number	1	15,178
<b>MODAL</b>	Modal verb	1	14,115
<b>POS</b>	Genitive marker	1	5,247
<b>ORG</b>	Organization	1	1,958
<b>FW</b>	Foreign Word	1	803

## Noise

Typically, text data includes various types of noise in varying quantity. What precisely qualifies as noise and how much of it will be normalized or eliminated depends on the goal, resources, and researcher. A list can be created which dictates the parameters of what can be included as POS. All tokens that are or comprise any symbol not listed above can be considered as noise.

Why is determining what is noise important? People are typically not interested in predicting tags for symbols, but only for what is typically considered as content. Another point is processing noise takes time and resources. Removing noise first speeds up the process.

### *johnIsAFireman.txt*

```
John is a Fireman in lower Manhattan in New York City. John
was there at the Twin Towers on that day in September.
```

This text can be tagged in two distinct ways: PTB and Aggregated. These POS lists are also done before any other pre-processing such as a Generalization Thesaurus so New, York, and City aren't all tagged individually.

### *PTB Tagging*

```
John/NNP is/VBZ a/DT Fireman/NN in/IN lower/JJR manhattan/NN
in/IN New/NNP York/NNP City/NNP ./ John/NNP was/VBD there/RB
at/IN the/DT Twin/JJ Towers/NN on/IN that/DT day/NN in/IN
September/NNP ./.
```

The aggregated tagging combines many PTB tags into one. In PTB is/VBZ and was/VBD are combined and both are tagged as /VERB.

### *Aggregated Tagging*

```
John/AGENTLOC is/VERB a/IRR Fireman/NOUN in/IRR lower/ADJ
manhattan/NOUN in/IRR New/AGENTLOC York/AGENTLOC City/AGENTLOC
./ John/AGENTLOC was/VERB there/IRR at/IRR the/IRR Twin/ADJ
Towers/NOUN on/IRR that/IRR day/NOUN in/IRR September/AGENTLOC
./.
```



## **Description**

Removing unwanted, or unneeded, items is important to reduce the amount of data to analyze. This include white space, punctuation, numbers and symbols.

## 🗑️ Remove Extra White Spaces

In many texts there are extra white spaces inserted between words or before and after punctuation. This is partly a holdover from the days of the typewriter of double spacing after a period. But with proportional fonts it's now an unnecessary practice. AutoMap finds instances of multiple spaces and replaces them a single space. Below is a text with varying number of white spaces between words.

### *spaces.txt*

```
one space. two  spaces. three   spaces. four    spaces.
```

After AutoMap removes extra white spaces it's much easier to read.

```
one space. two spaces. three spaces. four spaces.
```

## 🗑️ Remove Punctuation

Punctuation is mainly for use in making sure the reader understands how the words are expressed. During analysis they are somewhat unnecessary as the words themselves are more important. The Remove Punctuation function removes the following punctuation from the text: `.,:;' ">()!?-`. AutoMap will remove the punctuation and either close up the sapce between or insert a white space as a placemaker.

### *punctuation.txt*

```
"English" is hard (so very hard)!?! What's with all these  
commas (,), semi-colons (;), and colons (:).
```

### *Removing Punctuation and inserting white space*

```
English is hard so very hard What s with all these  
commas      semi colons      and colons
```

### *Removing Punctuation and NOT inserting white space*

```
English is hard so very hard Whats with all these  
commas semicolons and colons
```

## 💰 Remove Symbols

Symbols are parts of language which are not concepts but assist in understanding the language. Occasionally these symbols need to be removed to make semantic networks and meta-networks better understood.

## Removing the Default List of Symbols

AutoMap has a default list of symbols that can be removed:

```
~`@#$$%^&* _+={}[]\|/;<>.
```

**NOTE:** This option is an all-or-nothing function.

### *symbols.txt*

```
As he emailed {bob@jewelry.com} he knew the $200.00*
|+shipping| on [http://jewelry.com\~necklace] would = a ^50%
was a <`bargain>. And his #1 girl & mom deserved the best.
```

### *Removing Symbols and inserting white space*

```
As he emailed bob jewelry.com he knew
the 200.00 shipping on http: jewelry.com necklace wou
ld a 50 was a bargain . And his 1 girl mom deserved
the best.
```

### *Removing Symbols and NOT inserting white space*

```
As he emailed bobjewelry.com he knew the 200.00 shipping on
http:jewelry.comnecklace would a 50 was a bargain. And his 1
girl mom deserved the best.
```

## Removing a User Set of Symbols

The second option for removing symbols is to define the list you want removed. The list consists of one line with all the symbols to remove together with no spaces between the entries.

### *symbols.txt*

```
As he emailed {bob@jewelry.com} he knew the $200.00*
|+shipping| on [http://jewelry.com\~necklace] would = a ^50%
was a <`bargain>. And his #1 girl & mom deserved the best.
```

## Removing Set of Symbols containing {}[]#

As he emailed bob@jewelry.com he knew the \$200.00\* |+shipping| on http://jewelry.com/~necklace would = a ^50% was a <`bargain>. And his 1 girl & mom deserved the best.

## Remove Numbers

Removing numbers will remove not only numbers as individual concepts but also removes numbers embedded within concepts.

## Remove Options

There are two options for removing numbers.

1. Replacing the number(s) with a space
2. Removing the number(s) and closing the distance between the letters before and after.

## Examples

Remove numbers as individual concepts.

### *buckleMyShoe.txt*

1, 2, buckle my shoe! 3, 4, shut the door

#### *Text after RemoveNumber:*

, , buckle my shoe! , , shut the door.

Numbers within other concepts and closing up distance.

### *c3pO.txt*

C3PO was a robot in the movie Star Wars.

#### *Text after RemoveNumber:*

CPO was a robot in the movie Star Wars.

Numbers within other concepts and inserting white space.

### *c3pO.txt*

```
C3PO was a robot in the movie Star Wars.
```

### *Text after RemoveNumber:*

```
C PO was a robot in the movie Star Wars.
```



## Description

When processing data it's important to consider the order which preprocessing functions are done. In some circumstances the output will not be what you expect.

## Delete List and Generalization Thesaurus

In the example sentence the concept **the** is both as a stand alone concept and also as part of a title. The first instance is noise and can be eliminated but the second instance is part of the movie title.

### *rings.txt*

```
Dave likes the movie The Lord of the Rings
```

So you create a Delete List and a Generalization Thesaurus to remove the unwanted concepts but conserve the movie title.

### *ringsDeleteList.txt*

```
the  
of
```

### *ringsGenThes.csv*

```
The Lord of the Rings,The_Lord_of_the_Rings
```

## Run the Delete List then Thesaurus

If the Delete List is applied first with a rhetorical adjacency the following is obtained. You can see that the title can no longer be replaced by the Generalization Thesaurus.

Dave likes xxx movie xxx Lord xxx xxx Rings.

The replacement in the Generalization Thesaurus is impossible to apply as the **of** and the **the** in the title have been deleted.

## Run the Thesaurus then Delete List

But if the Generalization Thesaurus is applied first the title is replaced before the Delete List removes the noise.

Dave likes the movie The\_Lord\_of\_the\_Rings.

Then the Delete List can remove the other **unwanted** concepts.

Dave likes xxx movie The\_Lord\_of\_the\_Rings.



## Description

Stemming is a process for removing the more common morphological and inflectional endings from words in English. It detects inflections and derivations of concepts in order to convert each concept into the related morpheme. This assists in counting similar concepts in the singular and plural forms (e.g. plane and planes would normally be considered two terms). After stemming, "planes" becomes "plane" and the two concepts are counted together.

This can be broken down into two subclasses, **Inflectional and Derivational**.

- **Inflectional** morphology describes predictable changes a word undergoes as a result of syntax (the plural and possessive form for

nouns, and the past tense and progressive form for verbs are the most common in English). These changes have no effect on a word's **part-of-speech** (a noun still remains a noun after pluralizations).

- **Derivational** morphology may or may not affect a word's meaning (e.g.; '-ise', '-ship'). Although English is a relatively weak morphological language, languages such as Hungarian and Hebrew have stronger morphology where thousands of variants may exist for a given word. In such a case the retrieval performance of an IR system would be severely impacted by a failure to deal with such variations.

## K-STEM

**KSTEM or Krovetz stemmer** (Krovetz, 1995, a dictionary-based stemmer) : The Krovetz Stemmer effectively and accurately removes inflectional suffixes in three steps, the conversion of a plural to its single form (e.g. '-ies', '-es', '-s'), the conversion of past to present tense (e.g. '-ed'), and the removal of '-ing'. The conversion process firstly removes the suffix, and then though a process of checking in a dictionary for any recoding (also being aware of exceptions to the normal recoding rules), returns the stem to a word. This Stemmer is frequently used in conjunction with other Stemmers, making use of the advantage of the accuracy of removal of suffixes by this Stemmer. For the Krovetz stemmer, several customization options are offered:

## K-STEM Example

*tedInUSA.txt*

Ted lives in the United States of America. He lives on a dairy farm. He considers it a good life. Would he ever consider leaving?

*Text after K-Stemming:*

Ted live in the Unite State of America. He live on a dairy farm. He consider it a good life. Would he ever consider leave?

## Porter Stemming

The **Porter stemmer** uses the Porter Stemming algorithm. Additionally, it converts irregular verbs into the verb's infinitive.

## Porter Example

*tedInUSA.txt*

```
Ted lives in the United States of America. He lives on a dairy
farm. He considers it a good life. Would he ever consider
leaving?
```

*Text after Porter Stemming:*

```
Ted live in the Unite State of America. He live on a dairi
farm. He consid it a good life. Would he ever consid leav?
```

## Languages for Porter Stemming

Each language's stems work differently. Failing to use the correct language files when stemming risks obtaining incorrect results.

## Differences in Stemming

There is a difference in the way the Porter and K-Stem functions stem words: **consider(s) and dairy**.

**Porter** removes both the **er** and the **ers** from the words consider and considers. **K-Stem** removes the **s** from considers and both words end up as consider.

**Porter** changes the **y** in dairy to an **i** whereas **K-Stem** leaves the word untouched.

## Stem Capitalized Concepts

Decide whether or not to stem capitalized words. This will include all proper nouns.

**NOTE :** *If capitalized words are not stemmed then remember that the first word of each sentence will likewise not be stemmed.*

---

Porter, M.F. 1980. *An algorithm for suffix stripping*. I 14 (3): 130-137.

Krovetz, Robert 1995. *Word Sense Disambiguation for Large Text Databases*. Unpublished PhD Thesis, University of Massachusetts.



## Description

Outputs information regarding the currently loaded files. AutoMap writes one file for each file currently loaded.

### *milkAndCookies.txt*

```
Dave wants milk and cookies. He drives to the store. He then  
buys milk and cookies.
```

### *milkAndCookies.csv*

```
Number of Characters,83  
Number of Clauses,3  
Number of Sentences,3  
Number of Words,16
```



## Description

**Thresholds** refine the number of concepts to be included when creating the Union Concept List and the individual Concept List files. As the Threshold number is increased, concepts with frequencies less than the threshold are removed from the file when it is written.

## Example Texts

Below are three small text files. They are small for demonstration purposes. As will be seen, even small text repositories can create large Concept List files.

**theboy-1.txt** : See the boy named Dave. He has two toys. One toy is red and the other toy is blue.

**theboy-2.txt** : On Monday Dave plays with the blue toy. It's his favorite toy.

**theboy-3.txt** : On all other days Dave plays with the red toy.

## Global Threshold

Using the Global Threshold you can control which concepts will not be included in the Union Concept List. Any concept appearing less than the threshold will not be included in the Union Concept List file that's output.

First create a **Union Concept List** using the unprocessed text files. In large text files this can result in an unwieldy list.

## ucl.csv with no pre-processing

```
Words,Frequency,Relative Frequency,Relative Percentage
all,1,0.2,0.024390243902439025
and,1,0.2,0.024390243902439025
blue,2,0.4,0.04878048780487805
boy,1,0.2,0.024390243902439025
dave,3,0.6,0.07317073170731707
days,1,0.2,0.024390243902439025
favorite,1,0.2,0.024390243902439025
has,1,0.2,0.024390243902439025
he,1,0.2,0.024390243902439025
his,1,0.2,0.024390243902439025
is,2,0.4,0.04878048780487805
it's,1,0.2,0.024390243902439025
monday,1,0.2,0.024390243902439025
named,1,0.2,0.024390243902439025
on,2,0.4,0.04878048780487805
one,1,0.2,0.024390243902439025
other,2,0.4,0.04878048780487805
plays,2,0.4,0.04878048780487805
red,2,0.4,0.04878048780487805
see,1,0.2,0.024390243902439025
the,4,0.8,0.0975609756097561
```

```
toy,5,1.0,0.12195121951219512
toys,1,0.2,0.024390243902439025
two,1,0.2,0.024390243902439025
with,2,0.4,0.04878048780487805
Total,41
Mean,1.64
StDev,0.0
```

With these three short files the list is already unwieldy. To decrease the number of concepts, use pre-processing on the raw text using the Delete List, Stemming, and Thresholds

## Removing contractions

Notice the text contains the contraction **it's**. In other texts there will probably be many more. Use a thesauri during pre-processing to expand all contractions. This will expand **it's** to **it is** as well any other contractions found in the thesauri file.

## Removing plurals

Next we want to combine the concepts of **toy** and **toys**. They both reference the same item and should be counted as the same concept. Run **Stemming** using KSTEM.

## Running a Delete List

Use the Concept List Viewer to create a Delete List of unneeded concepts. Then apply this Delete List.

## The Revised Union Concept List

Now generate another concept list.

You will find a list of all the **non-deleted concepts**.

```
Words,Frequency,Relative Frequency,Relative Percentage
all,1,0.16666666666666666,0.030303030303030304
be,2,0.3333333333333333,0.06060606060606061
blue,2,0.3333333333333333,0.06060606060606061
```

```

boy,1,0.16666666666666666,0.030303030303030304
dave,3,0.5,0.09090909090909091
day,1,0.16666666666666666,0.030303030303030304
favorite,1,0.16666666666666666,0.030303030303030304
has,1,0.16666666666666666,0.030303030303030304
is,1,0.16666666666666666,0.030303030303030304
it,1,0.16666666666666666,0.030303030303030304
monday,1,0.16666666666666666,0.030303030303030304
name,1,0.16666666666666666,0.030303030303030304
one,1,0.16666666666666666,0.030303030303030304
other,2,0.33333333333333333,0.06060606060606061
play,2,0.33333333333333333,0.06060606060606061
red,2,0.33333333333333333,0.06060606060606061
see,1,0.16666666666666666,0.030303030303030304
toy,6,1.0,0.18181818181818182
two,1,0.16666666666666666,0.030303030303030304
with,2,0.33333333333333333,0.06060606060606061
Total,33
Mean,1.65
StDev,0.0

```

*There's a definite difference between the two lists. Originally there were 25 individual concepts. Now there's a total of 20. Using thresholds will reduce them even further.*

## Thresholds: Local=1 and Global=2

Now the list can be further refined by setting the **Local and Global threshold** parameters.

First, leave **Local to 1** but change **Global to 2**. This tells AutoMap that a concept must appear a total of two or more times in **all** text files to be included in the Union Concept List.

Create a new concept List.

```

Words,Frequency,Relative Frequency,Relative Percentage
be,2,0.33333333333333333,0.09523809523809523
blue,2,0.33333333333333333,0.09523809523809523
dave,3,0.5,0.14285714285714285
other,2,0.33333333333333333,0.09523809523809523
play,2,0.33333333333333333,0.09523809523809523
red,2,0.33333333333333333,0.09523809523809523
toy,6,1.0,0.2857142857142857
with,2,0.33333333333333333,0.09523809523809523
Total,21

```

Mean, 2.625  
StDev, 0.0

*The origin list contained 25 concepts. After pre-processing it contained 20 concepts. After setting the Global Threshold to 2 it now contains 8 concepts.*

Raising the Global threshold to 3 would remove `be`, `blue`, `other`, `play`, `red`, `and` `with` leaving only 2 concepts (`dave` and `toy`) in the file.

## Local Threshold

The Local Threshold works on individual files. As the threshold is raised, more concepts are removed from the individual concept list files.

Setting the **Local Threshold=2** and the **Global Threshold=1** will remove any concept that appears only once in any of the loaded files.

## The results of all three Runs

File	Total number of Concepts in Original File	Concepts written to files using Local Threshold=2
ucl-1.txt	12	2
ucl-2.txt	9	1
ucl-3.txt	8	0

## Example of Concept List per Text for ucl-1.txt

```
Words, Frequency, Relative Frequency, Relative Percentage  
be, 2, 0.6666666666666666, 0.4  
toy, 3, 1.0, 0.6  
Total, 5  
Mean, 2.5  
StDev, 0.0
```



## Description

Unioning files/networks is a way of combining two or more files/networks into a single unit. There are multiple ways to union a file or network and each will give differing results.

## Union Examples

Let's say for example that the terms **John** and **Mary** both appear in two separate files. Now let's say that in file 1 they are connected three times (frequency=3). And in the second file they are connected nine times (frequency=9).

### Minimum

The Minimum union of John and Mary will be the lowest number of connections in either file. In this example a frequency of 3 from file 1 becomes the result.

### Maximum

The Maximum union of John and Mary will be the highest number of connections in either file. In this example a frequency of 9 from file 2 becomes the result.

### Sum

The Sum union of John and Mary will be a total of all the frequencies added together. In this example file 1 frequency=3 and file 2 frequency=9. The sum of these two is 12.

### Average

The Average union of John and Mary will be the sum of the two frequencies divided by the total number of files used. In this example file 1 frequency=3 and file 2 frequency=9. The sum of these two is 12. Next divide this sum (12) by the number of files (2) and the result is 6.



## Window Size

### Description

The window size determines the span in which connections will be made. The larger the window size, the more connections within that window.

A connection is made between each concept within a window. The window will then shift one concept in the direction of the text (for instance, the window shifts right for most Latin-based languages) and create a new window to analyze. This will continue to the end of the text.

### Example

*cookiesAndMilk.txt*

I have cookies and milk

*Window of concepts 1-3 : I have cookies*

I have, I cookies, have cookies

*Window of concepts 2-4 : have cookies and*

have cookies, have and, cookies and

*Window of concepts 3-5 : cookies and milk*

cookies and, cookies milk, and milk

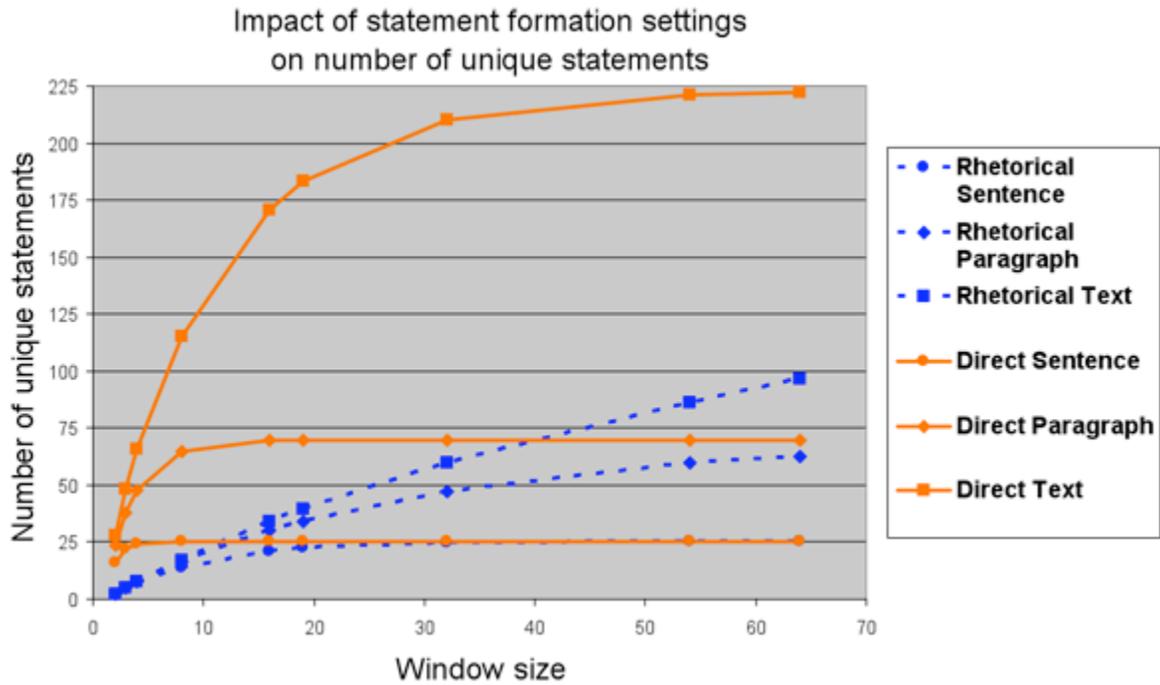
### Correct Window Size

Determining a correct window size is important. Choosing too small a window size may result in important links being missed. Too large a window size connects too many concepts, overwhelming important links.

Dave likes milk and cookies but John likes cauliflower

The example sentence above contains nine concepts. Manually reviewing this sentence reveals that milk and cookies are associated with Dave and cauliflower is associated with John.

But using a direction of **unidirectional** and a window size of **9** results in cauliflower also being associated with Dave.



## Description

This section contains general information about Semantic Networks and MetaNetworks.



# Description

AutoMap is concerned with a variety of different types of networks. Below is a chart showing the various types of networks and how they interact with each other.

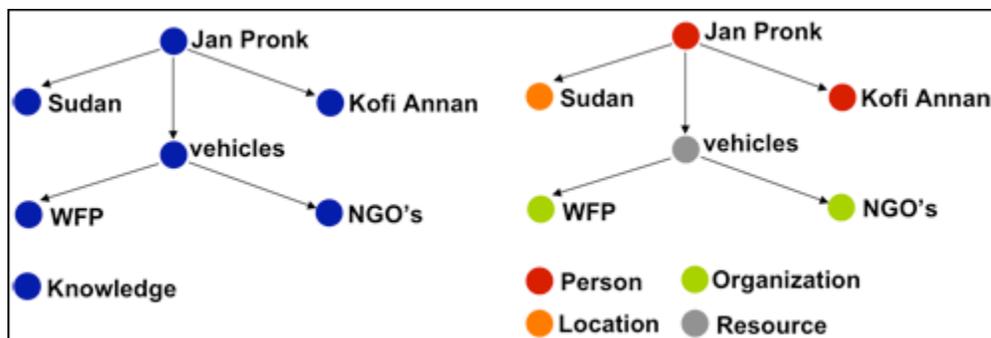
## Items it Detects:

<b>Agent</b>	<b>Interaction Network</b> Who knows who Structure	<b>Knowledge Network</b> Who knows what- Culture	<b>Assignment Network</b> Who is assigned to what-Jobs	<b>Employment Network</b> Who works where- Demography
<b>Knowledge</b>		<b>Information Network</b> What informs what-Data	<b>Requirements Network</b> What is needed to do what-Needs	<b>Competency Network</b> What knowledge is where- Culture
<b>Tasks</b>			<b>Precedence Network</b> What needs to be done before what- Operations	<b>Industrial Network</b> What tasks are done where-Niche
<b>Organizations</b>				<b>Inter-organizational Network</b> Which organizations work with which- Alliances

Meta-	Age	Knowled	Resourc	Task/Eve	Organizati	Location
-------	-----	---------	---------	----------	------------	----------

Network	nt	ge	e	nt	on	
agent	Social nw	Knowledge nw	capabilities nw	assignment nw	membership nw	agent location nw
Knowledge		Information nw	Training nw	Knowledge requirement nw	Organizational knowledge nw	Knowledge location nw
Resource			Resource nw	Resource requirement nw	Org. Capabilities nw	Resource location nw
Task/Event				Precedence nw	Org. Assignment nw	Task/Event nw
Organization					Interorg. nw	Org. location nw
Location						Proximity nw

## Network Types



## One Mode Network

Represent reality that people have in their minds and use to make sense of their surroundings.

Semantic Networks as Mental Models: Single Mode Networks are usually Semantic Networks. Nodes are not distinguished in any way. In the example

all nodes are classed as knowledge. Represent reality that people have in their minds and use to make sense of their surroundings. Cognitive constructs that reflect the subjects' knowledge and information about a certain topic.

## Multi-Mode Network

Identification and classification of all relevant instances of node and edge classes from texts as efficiently and accurately as possible.

Which agent or group is located where, has access to what resources, possesses what knowledge, is involved in what tasks, has what personal characteristics, ... ?

Nodes are classified by category and ORA can use these classifications for analysis. On the right nodes are classed as person, location, organization or resource.

Multi-Mode networks are **Ontologically coded** socio-technical networks which classify relevant nodes according to some ontology or taxonomy.



## Description

Semantic networks are knowledge representation schemes involving nodes and links between nodes. It is a way of representing relationships between concepts. The nodes represent concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph.

## Directional

**Uni-directional** : will only look forward in the text file for a relationship. Any concept that came before will be ignored.

**Bi-Directional** : will attempt to find a relationship in either direction of the concept. Both are constrained by windowSize and textUnit.

```
agent1 xxx xxx agent2 xxx xxx agent3.
```

Using **uni-directional** agent2 would have a relationship to agent3 but not agent1. Relationships can only look forward in the text.

Using **bi-directional** agent2 would have a relationship to both agent3 and agent1./p> Window Size

The distant concepts can be and still have a relationship to one another. Only concepts in same window can form statements. The window is defined in **textUnit**.

## Text Unit

The text unit can be comprised of one of the following:

**Sentence** : a sentence is a grammatical unit of one or more words.

**Word** : A word is a unit of language that represents a concept which can be expressively communicated with meaning

**Clause** : A clause consists of a subject and a verb. There are two types of clauses: independent and subordinate (dependent). An independent clause consists of a subject verb and also demonstrates a complete thought: for example, "I am sad." A subordinate clause consists of a subject and a verb, but demonstrates an incomplete thought: for example, "Because I had to move."

**Paragraph** : A paragraph is indicated by the start of a new line. It consists of a unifying main point, thought, or idea accompanied by supporting details.

**All** : The entire text

## Example

*dairyFarm.txt*

```
Ted runs a dairy farm. He milks the cows, runs the office, and  
cleans the barn.
```

### **Semantic Network parameters:**

```
windowSize="2" textUnit="S" directional="U" resetNumber="1"
```

### **Concept List:**

```
concept, frequency, relative_frequency, gram_type  
He,1,0.5,single  
Ted,1,0.5,single  
a,1,0.5,single  
and,1,0.5,single  
barn,1,0.5,single  
cleans,1,0.5,single  
cows,1,0.5,single  
dairy,1,0.5,single  
farm,1,0.5,single  
milks,1,0.5,single  
office,1,0.5,single  
runs,2,1.0,single  
the,3,1.5,single
```

### **Word List:**

```
Ted, runs, a, dairy, farm, He, milks, the, cows, runs, the,  
office, and, cleans, the, barn
```

### **Property List:**

```
Number of Characters,79  
Number of Clauses,4  
Number of Sentences,2  
Number of Words,16
```

### **Semantic Network csv:**

```
concept, concept, frequency  
He,milks,1  
Ted,runs,1  
a,dairy,1  
and,cleans,1  
cleans,the,1  
cows,runs,1  
dairy,farm,1  
farm,He,1  
milks,the,1  
office,and,1  
runs,a,1  
runs,the,1  
the,barn,1  
the,cows,1
```



## Description

The Meta-Network (Carley, 2002) Thesaurus maps key words in a text file with the categories to create a Meta-Network. This can be done at any step of the process but it is suggested that a Delete List and/or General Thesaurus is run previously. This makes sure that unnecessary terms aren't mapped into the network.

It is primarily used for preparing a file for importing into ORA and the creation of a semantic network to analyze. ORA looks for Nodes and NodeSets. This process groups those concepts into the NodeSets used by ORA.

A Meta-Network Thesaurus associates concepts with the following meta-network categories: Agent, Knowledge, Resource, Task/Event, Organization, Location, Action, Role, Attribute, Any user-defined category (as many as the user defines).



## Description

The **DyNetML Network Viewer** allows you to view a DyNetML files properties and relationships. From the pull-down menu select **Tools => DyNetML Network Viewer**. From the viewer's pull-down menu select **File => Open File**. Navigate to the xml file to view and click

**NOTE :** *This viewer will open any XML file. It will ignore attempts to open other types of files.*

The DyNetML viewer can examine both your semantic network files and your DyNetML files. Each file will display its structure and the individual properties of the nodes and networks.

## GUI

Each section will contain either a **+** or **-** button which will expand or contract that section.

## Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

**NOTE :** *The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.*

## Pull-Down Menus

### File Menu



**Open File :** Opens either Semantic or MetaNetwork files and display the file structure.



**Save As :** You can save the current network to a new directory under a new name.



**Exit :** Exits the DyNetML Viewer and returns to the Main GUI.

### View Menu

**Expand** : Expands out the entire network.

**Collapse** : Collapses the entire network.

## Procedures Menu



**Add Attribute:**



**Add Attributes:**



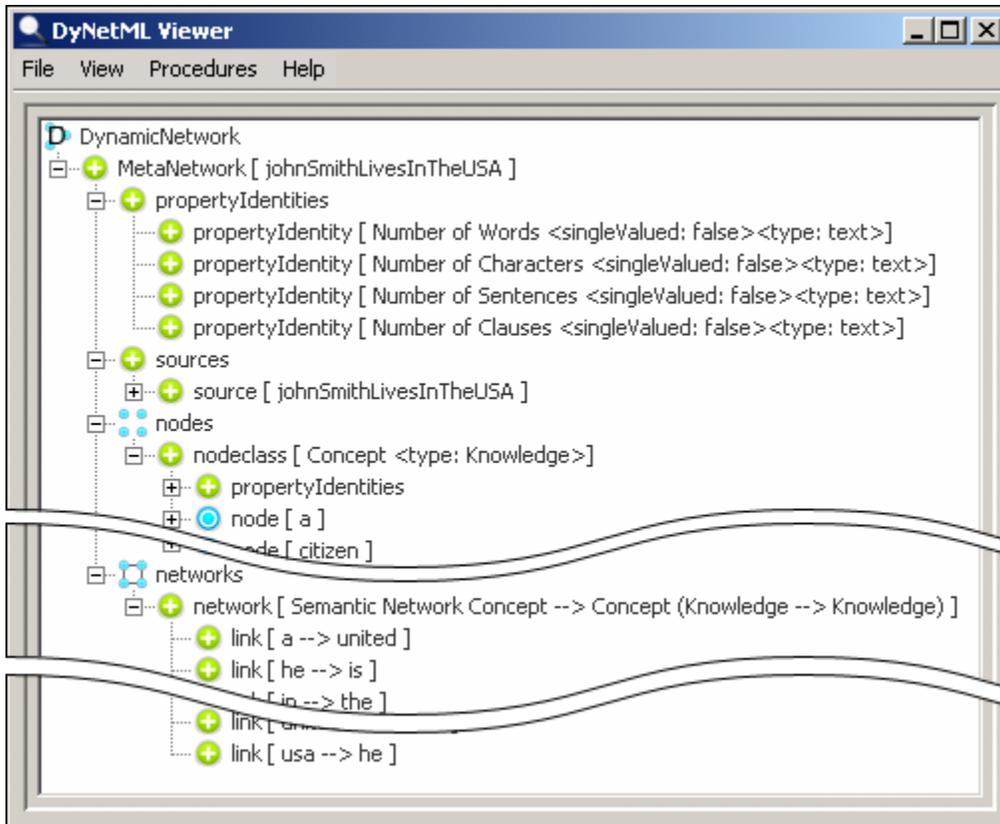
**Relocate Source Location:**



**Add Icon Reference to DyNetML:**

## Network Displays

### Displaying a Semantic Network



When viewing a Semantic Network the viewer will display four main areas:

**propertyIdentities**

Information about the source file, number of words, characters, sentences, and clauses.

**sources**

The source files in the semantic network

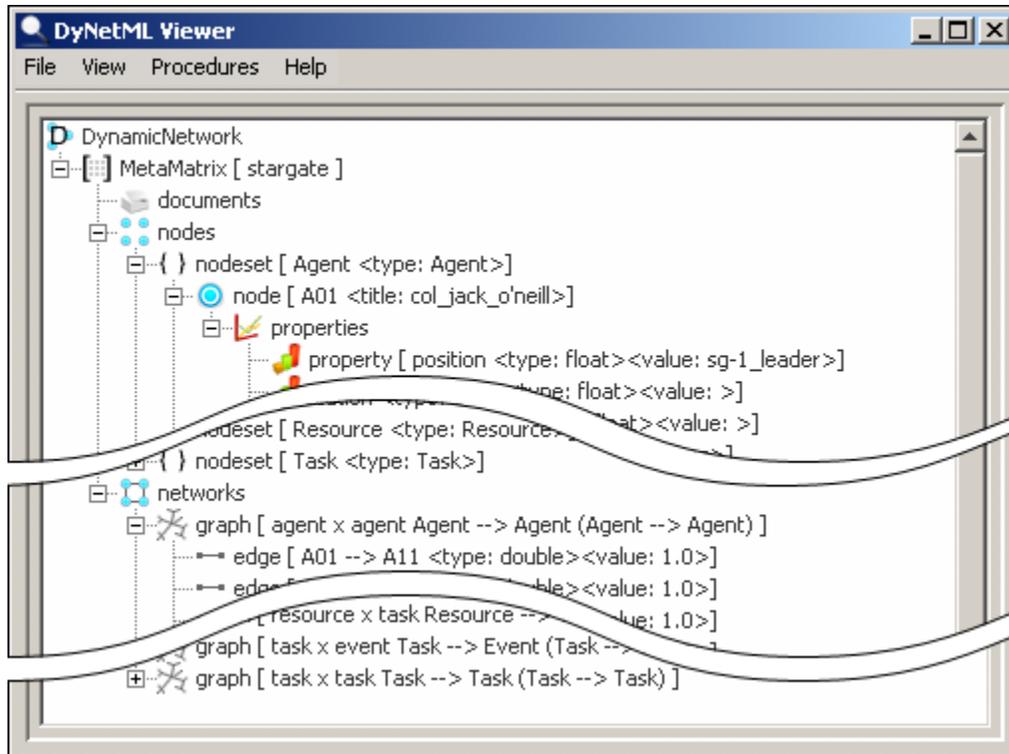
**nodes**

The nodeclasses in the semantic network and information regarding each nodeclass and node.

**networks**

Information on each network and the links contained in each network.

## Displaying a NetaNetwork



When viewing a Meta-Network (Carley, 2002) the viewer will display two main areas: **nodes and networks**.

### nodes

The nodeclasses and the nodes each contains and the properties of each node.

### networks

The graphs which make up each network and all the links contained in each network.



## Extracting a Semantic Network

## Description

Text files have connections but they are sometimes difficult to see. You can use AutoMap and process them to create semantic networks which can be viewed in ORA.

This lesson details processing text files in AutoMap to extract a Semantic Network, how to view it in ORA. Other lessons will detail specific reports that can prove useful.

## What is a Semantic Network?

Semantic networks are knowledge representation schemes involving nodes and links between nodes. It is a way of representing relationships between concepts. The nodes represent concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph.

## Procedure

This lesson will use the file: **JC\_summary-1.txt**.

## Load text document into AutoMap

Place all the text files for conversion into a single folder. From the Pull Down Menu select **File => Import Text Files**. The first text will be displayed in the main window and the filename will appear in the Filename Box. Using the File Navigation Buttons you can navigate through the loaded files.

## Build a General Thesauri

Many people, places and things are made up of two or more words. For example Julius Caesar, Brutus's House, status of Caesar. Before producing any files usable in ORA it's necessary to combine these multi-word concepts into **key concepts**.

**NOTE :** *Some concepts include the definite article in their name and should be included in the thesaurus.*

If you have no previous thesaurus then one will need to be created from scratch. This will require going through the text files and finding those multi-word concepts and creating a list of key concepts. The format for this is `multi word concept,key_concept`.

**NOTE :** Be sure **NOT** to leave any spaces before or after the comma.

Below is part of the Generalization Thesaurus that is used for this lesson. It contains concepts from the Julius Caesar text.

### *juliusCaesar-GenThes.csv*

```
Ides of March,Ides_of_March
Julius Caesar,Julius_Caesar
Julius Caesar's,Julius_Caesar
Julius Caesar's status,statue_of_Julius_Caesar
kill Caesar,kill_Caesar
kills herself,commit_suicide
king,emperor
letter,forged_letters
```

## Apply a General Thesauri

After the thesaurus is created it is time to apply it to the text. From the Pull Down Menu select **Preprocess => Apply Generalization Thesauri**. Navigate to the directory where the thesauri is saved and click **[Select]**. Next a dialog box will appear asking if you want to use **Thesaurus Content Only**. Leave the response as **No**.

See [Content => Thesaurus Content Only](#) for more information.

### Notes about Thesaurus Building:

1. In large texts there may be multiple person with the same first name.
2. The definite article in the concept like **the USDA** would be placed in the Thesaurus instead of being deleted in the Delete List.

## Create the Concept Lists

Next we need to create a Delete List. One way is to first create a Concept List and use this to help in creating a Delete List. The frequency attribute will assist in finding unneeded and unwanted terms.

From the main menu select **Generate => Concept List => Concept List (Per Text)**. Navigate to the directory to save the files and click **[Select]**. AutoMap will ask if you want to create a **Union Concept List**. Click **[No]** as you only have one file loaded.

**NOTE :** *With multiple files loaded you would select **Generate => Concept List => Concept List (Union Only)**. This creates one list for all files currently loaded.*

## Build a Delete List

Open the Concept List Viewer by selecting **Tools => Concept List Viewer**. From the viewer menu select **File => Open File**. Now navigate to the directory containing the newly created Concept List and click **[OK]**.

Click the header **Frequency**. This will sort the concepts by the number of occurrences in the file(s). To build a Delete List place a check mark in the **Selected** column for all the concepts you wish to place in the Delete List. When you are finished select **File => Save Delete List**. Navigate to the folder you want to save the Delete List file. Close the Viewer.

## Apply a Delete List

From the main menu select **Preprocess => Apply Delete List**. Navigate to the directory with your newly created file and click **[OK]**. You will be asked whether you want Rhetorical (replaces deleted concepts with a placeholder **xxx**) or Direct (removes the concept entirely) adjacency. For this lesson I choose rhetorical.

**NOTE :** *The placeholder **xxx** will not output to the DyNetML file as a concept.*

## Create a DyNetML file

Now it's time to generate the DyNetML. From the Pull Down Menu select **Generate => Concept List => Concept Network (Per Text)** for separate DyNetML files or **Generate => Concept List => Concept Network (Union Only)** to create one file with concepts from all files. AutoMap will output **XML file(s)** usable directly in ORA. You will be directed to select the destination folder for these file(s).

**NOTE :** When processing multiple files and selecting the **Per Text** function AutoMap will ask if you want to create a Union of all Semantic Network files.

The DyNetML file(s) will contain one NodeClass of Concepts. After loading into ORA Nodes can be separated into individual NodeClasses and links can be created to form Networks.



## Description

This section contains general information about processing using scripts.



## Using AutoMap 3 Script

The AutoMap 3 script is a command line utility that processes a large number of files using a set of processing instructions provided in the configuration file. Following is a simple explanation of how to construct a configuration file.

Once the configuration file has been created, the Automap 3 Script is ready to use. The following is a brief on running the script.

1. Configure the **AutoMap 3 .config file** as necessary. (Tag explanations in next section). Be sure to include pathways to input and output directories and the name of the config file to use.

```
<Settings>
<AutoMap
  textDirectory="C:\My Documents\dave\project\input "
  tempWorkspace="C:\My Documents\dave\project\output "
  textEncoding="unicode" />
</Settings>
```

2. Navigate to where AutoMap is installed.

3. At the prompt type: **am3script newProject.config** (where newProject.config is the config file you built).
4. AutoMap 3 will execute the script using the .config file specified.

## For Advanced Users

It is possible to set the your PATH environmental variable to include the location of the install directory so that AM3Script can be used in any directory from the command line. Please note this is not recommended for users that have no experience modifying the PATH environmental variable.

## Placement of Files

It is suggested the user create sub-directories for input files and output files in within an overall directory. This assists in finding the correct files later and prevents AutoMap from overwriting previous files. The **input** directory is empty except for your text files. The **output** will contain the output from AutoMap. The **support** directory will contain your Delete Lists, Thesauri, and any other files necessary during the run.

```
C:\My Documents\dave\project\input  
C:\My Documents\dave\project\output  
C:\My Documents\dave\project\support
```

**NOTE :** *It's important when typing in pathways that they are correct or AutoMap will fail to run.*

## Script name

The script.config file can be named whatever you like but we do recommend keeping the .config suffix. This way if you can do multiple runs to the files in a concise order: `step1.config`, `step2.config`, `step3.config`....

## Pathways

Pathways used in attributes are always relative to the location of AM3Script, (e.g. `/some_files` uses a directory `some_files` below the directory AM3Script is located in. A full pathway always begins with the drive name e.g. `C:/` and follows the pathway down to the files.

**NOTE :** Both relative and absolute paths can be used for the configuration path. Relative traces a path from the location the config to the file it needs (e.g. `..\..\anotherDirectory/aFile`). Absolute traces a pathway from the root directory to the file it needs (`C:\\{pathway}\aFile`).

If given a non-existent pathway you will receive an error message during the run.

## Tag Syntax in AM3Script

There are two styles of tags in the AM3Script script. The first one uses a set of two tags. The first tag starts a section and the second tag ends the section. The second tag will contain the exact same word as the first but will have, in addition, a "/" appended after the word and before the ending bracket. This designates it as an ending tag. All the parameters/attributes pertaining to this tag will be set-up between these two tags. e.g.

`<aTag></aTag>`.

The second style is the self-ending tag as it contains a "/" within the tag. Any attributes used with this tag are contained within the tag e.g. `<aTag attribute="attributeName"/>`.

## Output Directory syntax (TempWorkspace)

Output directories created in functions under the `<PreProcessing>` tag will all be suffixed with a number designating the order they were performed in. If a function is performed twice, each will have a separate suffix i.e. `Generalization_3` and `Generalization_5` denotes a Generalization Thesauri was applied to the text in the 3rd and 5th steps. Using `thesauriLocation` different thesauri could be used in each instance. For all other functions outside `PreProcessing` there is no suffix attached.

**NOTE :** The output directories specified above are in a temporary workspace and the content will be deleted if the AM3Script uses this directory again in processing. It is recommended that the directory specified in the temp workspace be an empty directory. Also, for output that user wishes to keep from processing it is recommend to use the `outputDirectory` tag within the individual processing step.

## Example

```
<AddAttributes3Col attributeFile="C:\My
Documents\dave\project\support\attributeFile"
outputDirectory="C:\My Documents\dave\project\output" />
```

By using these tags it allows the user to specify where they want the individual processing step output to go. It also makes finding the location of the output files much simpler instead of looking through the contents of the TempWorkspace.

## AutoMap 3 System tags

### **<Script></Script>** *(required)*

This set of tags is used to enclose the entire script. Everything used by the script must fall between these two tags. The only line found outside these tags will be the declaration line for xml version and text-encoding information: `<?xml version="1.0" encoding="UTF-8"?>`

*Need a list of the encodings*

### **<Settings></Settings>** *(required)*

Used for the setting for the default directories for text and workspace. For AM3Script the tag is `<AutoMap/>`

**NOTE :** *Any of the parameters can use inputDirectory and outputDirectory to override the default file location. These pathways will be relative to the location of the AM3Script.*

### **<AutoMap />** *(Required)*

The `<AutoMap/>` tag contains default pathways used by all functions and the type of text encoding to use. Any function can override these pathways by setting inputDirectory and outputDirectory within it's own tag. The location of text files to process is contained in `textDirectory="C:\My Documents\dave\project\input"`. The location of the files that will be written to the output directory is in `tempWorkspace="C:\My Documents\dave\project\output"`. To specify the encoding method to use set `textEncoding="unicode"` (currently UTF-8 is the default. AutoMap uses UTF-8 for processing. Please make sure to set text encoding to your correct specification of your text.). **AutoDetect** will attempt to detect and convert your text over to UTF-8.

## <Utilities></Utilities> *(required)*

The <Utilities> tag contains the sections <PreProcessing>, <Processing>, and <PostProcessing>. All three sections need to be nested within the <Utilities> tag.

## AutoMap 3 Preprocessing Tags

### <PreProcessing></PreProcessing> *(required)*

These are utilities that modify raw text. The order the steps are placed in the file is the order they are performed. You can also perform any of these utilities multiple times. i.e. perform a <Generalization/>, then a <DeleteList/>, then another <Generalization/>. Each step's results will be written to a separate output directory.

If inputDirectory or outputDirectory are used with any of the following tags they will override the directory pathways in under <Settings>. (e.g. `textDirectory="C:\My Documents\dave\project\input"` and `tempWorkspace=" C:\My Documents\dave\project\output"`). A warning will be displayed for both cases.

### <RemoveNumbers />

This parameter accepts either `whiteOut="y"` or `whiteOut="n"`. A "y" replaces numbers with spaces i.e. C3PO => C PO. A "no" removes the numbers entirely and closes up the remaining text e.g. C3PO => CPO.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <RemoveNumbers whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
```

```
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <RemoveSymbols />

This parameter accepts either `whiteOut="y"` or `whiteOut="n"`. A "y" replaces symbols with spaces. A "no" removes the symbols entirely and closes up the remaining text. The list of symbols that are removed:

`~`@#$$%^&*+_+={}[|\|/<>.`

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <RemoveSymbols whiteOut="y" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <RemovePunctuation />

This parameter accepts either `whiteOut="y"` or `whiteOut="n"`. A "y" replaces punctuation with spaces. A "no" removes the punctuation entirely and closes up the remaining text. The list of punctuation removed is: `.,:;'`

`"(!)?-.`

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
```

```

<PreProcessing>
  <RemovePunctuation whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <RemoveExtraWhiteSpace />

Find instances of multiple spaces and replaces them a single space.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <RemoveExtraWhiteSpace />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <Generalization />

The Generalization Thesauri are used to replace possibly confusing concepts with a more standard form. e.g. a text contains both [United States](#) and [U.S.](#) The Generalization Thesauri could have two entries which replace both the original entries with [united\\_states](#).

If [useThesauriContentOnly="n"](#) AutoMap replaces concepts in the Generalization Thesauri but leaves all other concepts intact. If [useThesauriContentOnly="y"](#) then AutoMap replaces concepts but removes all other concepts from output file.

```

<Script>

```

```

<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <Generalization thesauriLocation="C:\My
  Documents\dave\project\support\genThes.csv"
  useThesauriContentOnly="y" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <DeleteList />

The Delete List is a list of concepts to remove from the text files before output file. Set `adjacency="d"`, for direct, removes the space left by deleted words. Remaining concepts now become "adjacent" to each other. Set `adjacency="r"`, for rhetorical, removes the concepts but inserts a spacer within the text to maintain the original distance between concepts.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <DeleteList adjacency="r" deleteListLocation="C:\My
  Documents\dave\project\support\deleteList.txt" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <FormatCase />

FormatCase changes the output text to either "lower" or "upper" case. If `changeCase="l"` then AutoMap will output all text in lowercase. `changeCase="u"` outputs all text in uppercase.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <FormatCase changeCase="u" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <Stemming />

Stemming removes suffixes from words. This assists in counting similar concepts in the singular and plural forms. i.e. plane and planes would normally be considered two terms. After stemming planes becomes plane and the two concepts are counted together.

`type="k"` KSTEM or Krovetz stemmer.

`type="p"` Porter Stemming.

The `kStemCapitalization="y"` tells AutoMap to stem capitalized words. `kStemCapitalization="n"` ignores capitalized words.

The `porterLanguage` parameter allows the user to select from various languages available. Currently the available languages are: **Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Portuguese, Russian, Spanish, and Swedish.**

**NOTE :** If you select Porter Stemming then a language **MUST** be chosen or the script will error.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
  <Stemming type="k" porterLanguage=""
  kStemCapitalization="y|n" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <Processing> (required)

These steps are performed after all **Pre-Processing** is finished. They are performed in the order they appear in the AM3Script.

## <POSExtraction />

`postType="ptb"` specifies a tag for each part of speech.

`postType="aggregate"` groups many categories together using fewer Parts-of-Speech tags.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
```

```
<posType="ptb" />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <Anaphora />

An anaphoric expression is one represented by some kind of deictic, a process whereby words or expressions rely absolutely on context. Sometimes this context needs to be identified. These definitions need to be specified by the user. Used primarily for finding personal pronouns, determining who it refers to, and replacing the pronoun with the name.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <posType="ptb" />
  <Anaphora />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

*For Anaphora to work POS must be run first.*

## <ConceptList />

Creates a list of concepts for each loaded text file. A Delete List or Generalization Thesauri can be performed before creating these lists to reduce the number of concepts in each file. These output files can be loaded into a spreadsheet and sorted by any of the headers.

```
<Script>
<Settings>
```

```

<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <ConceptList />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <SemanticNetworkList />

`windowSize="aNumber"` defines the distance between concepts which can have a relationship. `textUnit="S"`=sentence, "W"=word, "C"=clause, "P"=paragraph. "A"=all defines the units used. `resetNumber="aNumber"` defines the number of textUnits to process before resetting the window. `directional="U"` (unidirectional) looks forward in the text file only. `directional="B"` (Bi-Directional) finds relationships in either direction.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <windowSize="2 textUnit="S" resetNumber="2" directional="U" />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <MetaNetworkList />

This associates text-level concepts with Meta-Network (Carley, 2002) categories {agent, resource, knowledge, location, event, group, task, organization, role, action, attributes, when}. Concepts can be translated into several Meta-Network categories. `thesauriLocation="C:\My Documents\dave\project\thesauri"` designates the location of the MetaNetwork (Carley, 2002) Thesauri, if used.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <MetaNetwork thesauriLocation="C:\My
  Documents\dave\project\support\MetaNetworkThesauri.csv" />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <UnionConceptList />

Union Concept Lists consider concepts across all texts currently loaded, rather than only the currently selected text file. It reports total frequency, related frequency, and cumulative frequencies of concepts in all text sets. It's helpful in finding frequently occurring concepts over all loaded texts.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <UnionConceptList />
</Processing>
```

```
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

**NOTE :** *The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.*

## <NGramExtraction />

Extracts NGrams.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <NGramExtraction />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

## <CRFSuggestion />

This option automatically estimates mapping from text words from the highest level of pre-processing to the categories contained in the Meta-Network.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
```

```

<Processing>
  <CRFSuggestion />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <PostProcessing> (required)

The last step is adding in additions to the files with the PostProcessing functions. This includes adding attributes and Unionizing DyNetML files.

### <addAttributes />

Additional attributes can be added to the nodes within the generated DyNetML file. `attributeFile="C:\My Documents\dave\project\support"` is the pathway to the file which contains a header row with the attribute name.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <addAttributes attributeFile="C:\My
    Documents\dave\project\support\attributeFile.txt" />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

### <addAttributes3Col />

`<AddAttributes3Col attributeFile="C:\My Documents\dave\project\attributeFile" />` is similar to `<addAttributes>` but uses `name` and `value`.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <addAttributes3Col attributeFile="C:\My
  Documents\dave\project\support\3ColAttributeFile.txt" />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

```

## <UnionDyNetml />

UnionDyNetml creates a union of all dyNetml in a specified directory. It requires a unionType which can be `s` or `m`. `unionType="s"` is for a union of semantic networks and `unionType="m"` is for metanetworks.

```

<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input "
tempWorkspace="C:\My Documents\dave\project\output "
textEncoding="unicode" />
</Settings>
<Utilities>
<PreProcessing>
</PreProcessing>
<Processing>
  <UnionDyNetml unionType="s" />
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>

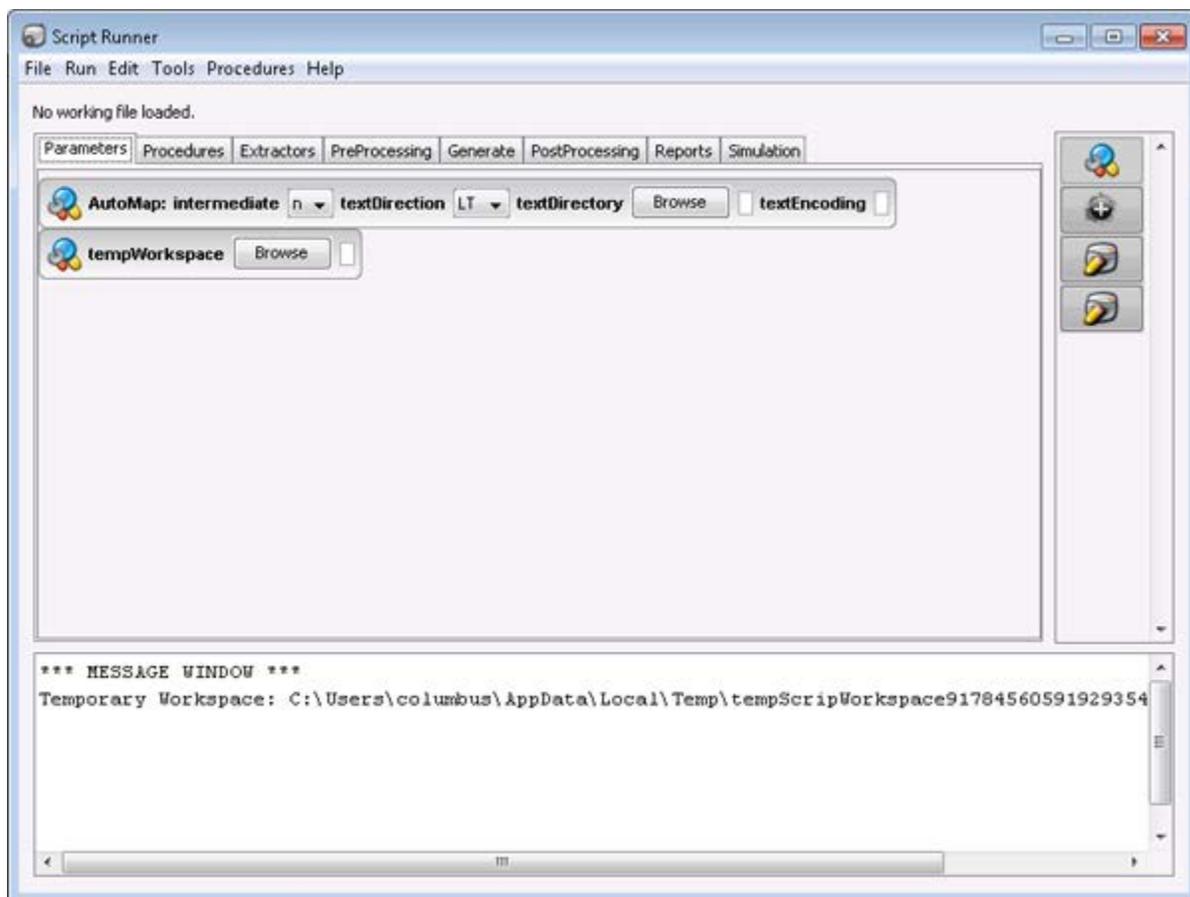
```



## Script Runner

Script Runner is explicitly used to process large sets of data from parameters tested from running a limited set of data in the GUI. After creating and modifying a set of functions in the GUI you can use those parameters to create you .aos file in order to process large sets.

And after a script is created and loaded again, many of the functions can be altered to obtain a different set of results (*e.g. change the Delete List run on a set of files*).



## GUI

The GUI consists of four parts. 1) The Menu; 2) The Tabs; 3) The Quick Launch buttons; and 4) The Message Window.

## Menus

## File Menu



**Load Script File** : Loads a script file either created in an external program or created previously in Script Runner.



**New Script File** : Create a New Script file from scratch



**Save** : Saves currently loaded script file



**Save As...** : Saves currently loaded script file that can be renamed as new file.



**Save Message Window Log** : As AutoMap is running your script it will display details on the actions it has performed. You can save these messages to a text file.

## Run



**Run This Script File** : Runs the script currently loaded in the viewer pane.



**Run This Script File as SuperScript** : Runs the script currently loaded in the viewer pane under multiple processors



**Script 2 BPEL** : Converts a file from ScriptRunner into a format usable by the SORASCS server.

## Edit



**Suggest Variables** :



**Suggest Temporary Directory** :



**Preprocess Script File** :



**Script 2 Package** :

# Tools

In addition to running scripts the Script Runner tool can call up other viewers. These can be used to verify the state of your files before or after running a script without leaving the viewer.

-  **Delete List Editor** : Calls the external Editor to work with a Delete List.  
See [Tools => Delete List Editor](#) for more information.
-  **Thesaurus Editor** : Calls the external editor to work with a Thesaurus file.  
See [Tools => Thesaurus Editor](#) for more information.
-  **Concept List Viewer** : Calls the external viewer to review a Concept List  
See [Tools => Concept List Viewer](#) for more information.
-  **Table Viewer** : Allows the user to view table files other than Concept Lists and DyNetML files.  
See [Tools => Table Viewer](#) for more information.
-  **XML Network Viewer** : Allows the user to view DyNetML and other XML.  
See [Tools => XML Viewer](#) for more information.
-  **Tagged Text Viewer** :  
See [Tools => XML Viewer](#) for more information.
-  **Script Config** :
-  **Add Plugin** :

# Procedures

-  **Run a Script File** : Navigate to the .config file you want to run. This can be a script you created in a text editor or a script created from AutoMap's main GUI pull-down menu **File => Save Script File** which will create a script of all current preprocessing steps.



**Run a Script File as SuperScript** : Allows user to run a script under multiple processors. User inputs the number of processors to use and AutoMap splits the input files into that many batches.

## Script Runner Tabs

The tabs at the top of the window are performed from left to right and all functions within a specific window are performed from top to bottoms. They include:

**Parameters** : Maintains information on the workspace and other information about the files being processed.

**Procedures** : Functions to prepare data files and support files which includes merging Delete Lists and Thesauri files.

**Extractors** : Used to get information from sources other than standard text files which includes FacebOok, Blogger, Twitter, and RSS feeds.

**PreProcessing** : Includes all the Preprocessing functions found in the GUI which includes Delete List, Thesauri, and various removal functions.

**Generate** : After all PreProcessing is finished these functions **generate** some type of output which includes Semanatic List, Meta-Networks, and other lists of concepts.

**PostProcessing** : Works on generated files to further process them which includes attributes, beliefs, and unions.

**Reports** Contains the reports useful after all processing is complete on text files.

**Simulation**

## Quick Launch Buttons

The set of buttons will change when a different tab is selected. The buttons will be functions needed for each different function.

## Message Window

Keeps track of all the user's actions and is also editable. In addition the message window can be saved.



## Description

All of AutoMap's functions can be accessed through the script. The two required files are the AM3Script (The AutoMap program) and a .config file (designed by the user). Additional files could include Delete Lists, Thesauri, or other list files necessary by the program.

## Create a Workspace

A good starting point is creating a project directory, a place where all your input (your text files), output (files AutoMap writes), and support (required files by certain functions) files will reside. This helps prevent files from getting lost. One suggestion is to create a top level project directory then create input, output, and support directories within that directory.

```
C:\My Documents\dave\project\input  
C:\My Documents\dave\project\output  
C:\My Documents\dave\project\support
```

## The .config file

Find the blank .config file in the AutoMap directory and make a copy. Rename this to something regarding your project. Open it in your text editor to begin editing the file. The blank .config file will appear as below.

```
<?xml version="1.0" encoding="UTF-8"?>  
<Script>  
<Settings>  
  AutoMap textDirectory="" tempWorkspace="" textEncoding="" />
```

```
</Settings>

<Utilities>
<PreProcessing>

</PreProcessing>
<Processing>

</Processing>
<PostProcessing>

</PostProcessing>
</Utilities>
</Script>
```

## Initial Setup

The first thing to do is tell AutoMap where your input files are and where you want the output files to be written.

```
<Settings>
  AutoMap textDirectory="C:\My Documents\dave\project\input"
  tempWorkspace="C:\My Documents\dave\project\output"
  textEncoding="" />
</Settings>
```

## PreProcessing Functions

Now decide which functions of AutoMap you need to run on your files. These are divided into three areas: **Preprocessing, Processing, and PostProcessing**. Review the documentation on the various functions to decide which functions you need to run on your text.

## A Generalization Thesaurus

Usually a Generalization Thesaurus is the first file to create. This can be done in either a text editor or spreadsheet. Create a list of single/multi word concepts from the text and the key concepts they should be translated to.

In a **text editor** create each pair on a single line separated by a comma. Make sure to **NOT** leave a space between the comma and the two items.

```
United States of America,United_States_of_America
```

Save this file as a **.csv** file.

In a **spreadsheet program** place the single/multi word concept in the first column and the key concept in the second column.

A	B
United States of America	United_States_of_America

Save this file as a **.csv** file.

In your project .config file in the Preprocessing section insert the command for applying a Generalization Thesaurus. Place the pathway to the newly created thesaurus in the **thesauriLocation** parameter and choose whether to use the **thesauriContentOnly** option.

**NOTE :** *thesauriContentOnly is set to **y** (put only concepts from the thesaurus in the output file) or **n** (use all concepts form the text files).*

```
<PreProcessing>  
  <Generalization thesauriLocation="C:\My  
  Documents\dave\project\support\thesauri.csv"  
  useThesauriContentOnly="y" />  
</PreProcessing>
```

Save the file.

## A Delete List

After all the key concepts have been identified it's time to find the **unneeded and unwanted** concepts. A Delete List removes these concepts and reduces the overall number of concepts to analyze. The procedure for applying a Delete List is similar to applying a thesaurus.

In a **text editor** create a list of concepts to be removed from the text. Each line should contain only one concept which consists of a single word. There should be no extra spaces or punctuation included.

```
and  
the  
but
```

Save this file as a **.csv** file.

In a **spreadsheet program** place each concept to delete in a single cell in the first column

A
and
the
but

Save this file as a **.csv** file.

In your project .config file in the Preprocessing section insert the command for applying a Delete List. Put the pathway to the newly created Delete List in the **deleteListLocation** parameter and choose whether to use the **saveTexts** option.

```
<PreProcessing>
  <Generalization thesauriLocation="C:\My
  Documents\dave\project\support\thesauri.csv"
  useThesauriContentOnly="y" />
  <DeleteList adjacency="r" deleteListLocation="C:\My
  Documents\dave\project\support\deleteList.txt" saveTexts="y" />
</PreProcessing>
```

Save the file.

## Other Preprocessing Functions

Any number of the preprocessing functions can be included in the script file in whatever order you need them. Insert the commands within the **<Preprocessing>** tags in the order you need them performed.

**NOTE :** *am3script will perform these function in the order they are placed in the script. Make sure you know what order you need to perform each task.*

## Processing Functions

Next thing to consider are the steps to take after the preprocessing is finished. These include most of the functions that output files and are based on the text after preprocessing. To run a Processing function the command is placed between the **<Processing>** tags.

## Processing functions with no parameters

Processing functions which take no parameters include **Anaphora**, **ConceptList**, **UnionConceptList**, and **NGramExtraction**. Placing these tags between the **<Processing>** tags automatically performs these functions in the order that order.

```
<Processing>
  <ConceptList />
  <UnionConceptList />
</Processing>
```

## Processing functions with parameters

All other Processing functions require parameters to set for successful completion. This is either the location of a support file or parameters necessary to complete the function. The parameters follow the tag and are separated by a space and the value is enclosed in quotes.

```
<Processing>
  <SemanticNetworkList directional="U" resetNumber="1"
  textUnit="S" windowSize="5" />
</Processing>
```

If a tag requires a file to work then the pathway needs to be placed in the location parameter. An incorrect pathway will cause the AutoMap function to fail.

```
<Processing>
  <Meta-Network thesauriLocation="C:\My
  Documents\dave\project\support\thesauri.csv" />
</Processing>
```

## PostProcessing Functions

The last step is to determine which, if any, postprocessing functions are needed. These are used to alter the DyNetML before running in ORA. They include **addAttribute**, **addAttribute3Col**, and **UnionDyNetml**. The **addAttribute** and **addAttribute3Col** both take an external file with a list of attributes to add. The **UnionDyNetml** requires the **unionType** to create either a **s** semantic or **m** Meta-Network (Carley, 2002) output file.

```
<PostProcessing>
  <addAttribute attributeFile="C:\My
  Documents\dave\project\support\attribute.csv" />
  <unionDynetml unionType="s" />
</PostProcessing>
```

Output files will be written to the designated output directory.

## Running the Script

After all the Preprocessing and Processing tags are completed the script can now be run.

Open a Command Window from the Start menu and navigate to the directory containing AM3Script. Move your **.config file** to this directory. At the command prompt type **am3script project.config**. Am3script will execute using the .config specified. Output files will be found in the directory specified in the tempWorkspace parameter.

```
<Settings>
  AutoMap textDirectory="" tempWorkspace="" textEncoding="" />
</Settings>
```

**NOTE :** Be sure to leave a space between *am3script* and the name of your config file.