AutoMap User's Guide 2009

Kathleen M. Carley, Dave Columbus, Mike Bigrigg, Jana Diesner, and Frank Kunkel

June 2009
CMU-ISR-09-114

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

Center for the Computational Analysis of Social and Organizational Systems
CASOS technical report.

*This report/document supersedes CMU-ISR-08-123
"Automap User's Guide 2008", July 2008*

This work was supported in part by the Office of Naval Research under Contract No. N00014-06-1-0772, ONR, and N00014-06-10921, by the National Science Foundation IGERT in CASOS, the Air Force Office of Sponsored Research with a MURI with George Mason University under Grant No. 600322GRGMASON, and the Army Research Lab under Grant No. DAAD19-01-2-0009. Additional support was provided by the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the Office of Naval Research, the Air Force Office of Sponsored Research, the Army Research Lab or the U.S.

Key Words: Semantic Network Analysis, Dynamic Network Analysis, Mental Modes, Social Networks, AutoMap

Abstract

AutoMap is software for computer-assisted Network Text Analysis (NTA). NTA encodes the links among words in a text and constructs a network of the links words. AutoMap subsumes classical Content Analysis by analyzing the existence, frequencies, and covariance of terms and themes.

Table of Contents

AutoMap 3 Overview 1

- An Overview 1
- Network Text Analysis (NTA) 1
- Semantic Network Analysis 1
- Social Network Analysis (SNA) 2
- Dynamic Network Analysis 2

Glossary 4

- Glossary 4

The GUI (Graphic User Interface) 11

- Description 11
- The GUI 11
- The Pull Down Menu 11
  - File 11
  - Edit 12
  - Preprocess 12
  - Generate 12
  - Tools 12
  - Help 12
- File Navigation Buttons 12
- Preprocess Order Window 12
- Filename Box 12
- Text Display Window 12
- Message Window 12
- Quick Launch Buttons 13

File Menu 13

- Description 13
  - Select Input Directory 13
  - Import Text 13
  - Save Preprocessed Text Files 14
  - Exit AutoMap 14

Edit Menu 14

- Description 14
  - Set Font 14

Preprocessing Menu 15
- Description 15
  - Undo 15
  - Remove Extra White Space 15
  - Remove Punctuation 15
  - Remove Symbols 15
  - Remove Numbers 15
  - Convert to Uppercase 16
  - Convert to Lowercase 16
  - Apply Stemming 16
  - Apply Delete List 16
  - Apply Generalization Thesauri 16

Generate Menu 16
- Description 16
  - Concept List 17
  - Semantic List 17
  - Parts of Speech 17
  - Semantic Network 17
  - MetaNetwork DyNetML 17
  - BiGrams 17
  - Text Properties 18
  - Named Entities 18
  - Feature Extraction 18
  - Suggested MetaNetwork Thesauri 18
  - Union Concept Lists 18

Content Section 19

Content 19

Anaphora 19
- Description 19
- Definition of Anaphora 19
  - Example 19
- What is NOT an anaphora 20

Bi-Grams 20
Description 20
Definitions 20
Threshold: 21
Thresholds Example 21
Bi-gram list 21
Bi-grams List using Delete List and Generalization Thesaurus 21
Bi-Gram Chart 22
Description 22
Concept Lists 24
Description 24
Example: 24
Delete Lists 25
Description 25
Points to Remember 25
Adjacency 26
Direct Adjacency 26
Rhetorical Adjacency 26
Reasons NOT to use a Delete List 27
Text Encoding 27
Description 27
Text Direction 28
Directionality 28
Feature Selection 29
Description 29
Date Styles 29
AutoMap understands certain styles of dates as shown below. 29
File Formats 30
Description 30
Other text formats 30
Format Case 32
Description 32
Example 32
Named Entities 33

Description 33
Items it Detects: 33
Networks 33
Description 33
Items it Detects: 33
Parts of Speech 35
Description 35
The Hidden Markov Model 35
Penn Tree Bank (PTB) Parts of Speech Table 35
Aggregate Parts of Speech 37
Aggregation of PTB Categories 37
Noise 37
Example 38
Process Sequencing 39
Description 39
Delete List and Generalization Thesaurus 39
Delete List 39
Generalization Thesaurus 39
Run the Delete List then Thesaurus 39
Run the Thesaurus then Delete List 39
Remove Numbers 40
Description 40
Remove Options 40
Examples 40
Remove Punctuation 41
Description 41
Example 41
Remove Symbols 42
Description 42
Example 42
Remove White Spaces 43
Description 43
Example 43
Semantic Lists 44

- Description 44
- Direction 44
- Window Size 44
- Text Unit 44

Semantic Networks 45

- Description 45
- Directional 45
- Text Unit 46
  - Example 47

Stemming 48

- Description 48
- K-STEM 48
  - K-STEM Example 49
- Porter Stemming 49
  - Porter Example 49
  - Languages for Porter Stemming 50
- Differences in Stemming 50
- Stem Capitalized Concepts 50

Text Properties 50

- Description 50

Thesauri, General 51

- Description 51
- Format of a Thesauri 51
- Uses for a Generalization Thesauri 51
  - Combining multi-word concepts 51
  - Normalizing abbreviations 52
  - Normalizing contraction 52
  - Correcting typos 52
  - Globalizing countries 53
- Example: 53
  - Example with ThesauriContentOnly not activated 53
  - Example using ThesauriContentOnly 54
- Stop Characters 54
- Why the Order of thesauri entries is Important 54

Thesauri, MetaNetwork 55
Description 55
Meta-Network categories 56
Example: 57
Thesaurus Content Only 58
Description 58
Thesaurus content only options: 58
Threshold, Global and Local 59
Description 59
Example Texts 59
Global Threshold 59
ucl.csv with no pre-processing 59
Removing contractions 60
Removing plurals 60
Running a Delete List 60
The Revised Union Concept List 61
Thresholds: Local=1 and Global=2 61
Local Threshold 62
The results of all three Runs 62
Example of Concept List per Text for ucl-1.txt 62
Union Concept List 63
Description 63
Definitions 63
Example 64
Using in Excel 66
Window Size 66
Description 66
Example 67
Correct Window Size 67
Tools 68
Concept List Viewer 68
Description 68
Sorting 69
Selecting Concepts 69

Compare Files 70
Create a Delete List 71
Delete List Editor 72
Description 72
Procedure 72
Semantic List Viewer 73
Description 73
Procedure 73
Script 75
Description 75
AM3Script 75
Using AutoMap 3 Script 75
For Advanced Users 76
Placement of Files 76
Script name 76
Pathways 77
Tag Syntax in AM3Script 77
Output Directory syntax (TempWorkspace) 77
Example 78
AutoMap 3 System tags 78
<Script></Script> (required) 78
<Settings></Settings> (required) 78
<AutoMap /> (Required) 78
<Utilities></Utilities> (required) 79
AutoMap 3 Preprocessing Tags 79
<PreProcessing></PreProcessing> (required) 79
<RemoveNumbers /> 79
<RemoveSymbols /> 80
<RemovePunctuation /> 80
<RemoveExtraWhiteSpace /> 81
<Generalization /> 81
<DeleteList /> 82
<FormatCase /> 82
<Stemming /> 83

<Processing> (required) 84
<POSExtraction /> 84
<Anaphora /> 84
<ConceptList /> 85
<SemanticNetworkList /> 86
<MetaNetworkList /> 86
<UnionConceptList /> 87
<NGramExtraction /> 87
<CRFSuggestion /> 88
<PostProcessing> (required) 88
<addAttributes /> 88
<addAttributes3Col /> 89
<UnionDynetml /> 89
DOS Commands 90
Description 90
CD: Change Directory 90
cd\ 90
cd.. 90
cd windows 91
cd\windows 91
cd windows\system32 91
cd 91
DIR: Directory 91
dir /ad 91
dir /s 92
dir /p 92
dir /w 92
dir /s /w /p 92
dir /on 92
dir /o-n 92
dir \ /s |find "i" |more 92
dir > myfile.txt 93

MD: Make Directory 93

md test 93

md c:\test 93

RMDIR: Remove Directory 93

rmdir c:\test 93

rmdir c:\test /s 93

COPY: Copy file 93

copy *.* a: 93

copy autoexec.bat c:\windows 94

copy win.ini c:\windows /y 94

copy myfile1.txt+myfile2.txt 94

copy con test.txt 94

RENAME: Rename a file 94

rename c:\chope hope 94

rename *.txt *.bak 94

rename * 1_* 94

AutoMap 3 Overview

An Overview

AutoMap is a software tool to analyze text using the method of Network Text Analysis. It performs a specific type of Network Text Analysis called Semantic Network Analysis. Semantic analysis extracts and analyzes links among words to model an authors **mental map** as a network of links. Additionally, Automap 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)

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

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). 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.

Social Network Analysis (SNA)

Social Network Analysis 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 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 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 (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). 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 comprise a network also.

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). 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 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 admit the analysis of multiple networks simultaneously in which, there are multiple types of entities (multi-entities) and multiple types of links (multiplex). 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.

DNA statistical tools tend to provide more measures to the user, because they have measures that use data drawn from multiple networks simultaneously. From a computer simulation perspective, entities in DNA are like atoms in quantum theory, entities 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: A company's employees can learn new skills and increase their value to the network; Or, kill one terrorist and three more are forced to improvise. Change propagates from one entity to the next and so on. DNA adds the critical element of a network's evolution and considers the circumstances under which change is likely to occur.

Glossary

Glossary

Adjacency Network : A Network that is a square actor-by-actor (i=j) network where the presence of pair wise 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. Therefore, it's the number of nodes who an node is connected to indirectly through their 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 (an 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 : File structure meaning Comma Separated Value. Common output structure used in database programs for formatting 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.

Endain : 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-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.

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.

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 same task, 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).

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

Thesauri : Associates concepts with more abstract 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*.

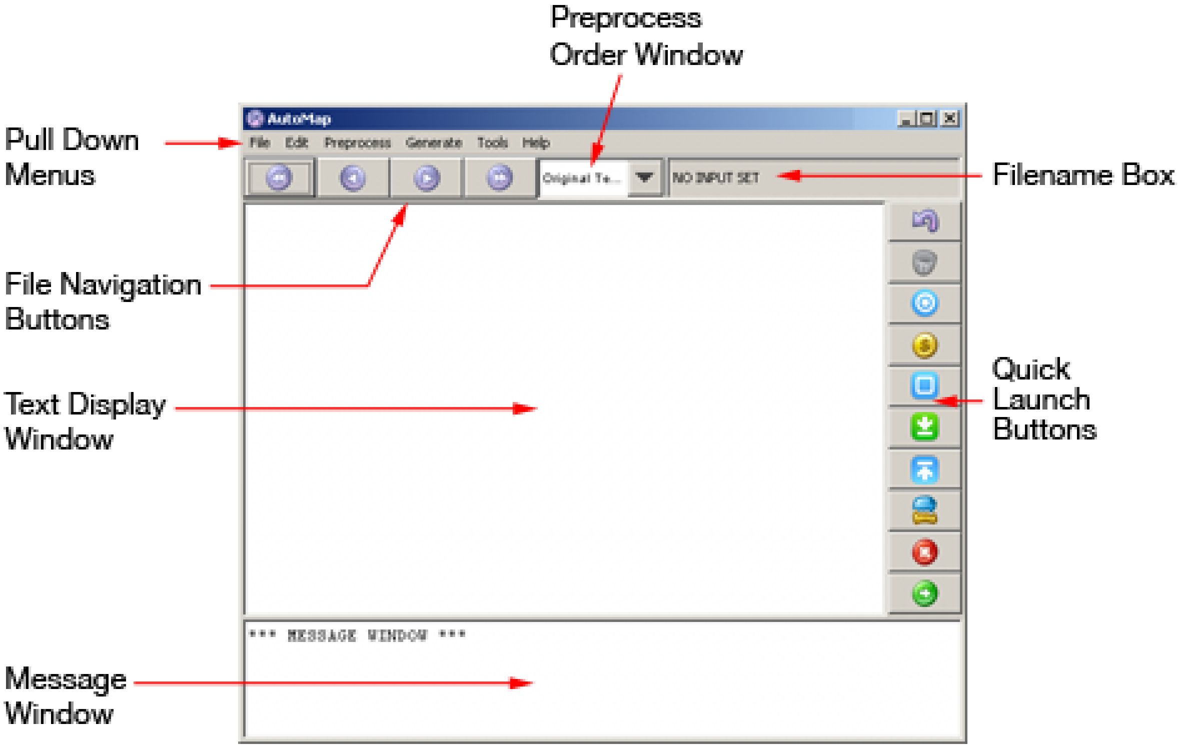
Windowing : A method that codes the text as a map by placing relationships between pairs of Concepts that occur within a window. The size of the window can be set by the user.

The GUI (Graphic User Interface)

Description

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

The GUI



```
*** MESSAGE WINDOW ***
```

The Pull Down Menu

File

Used for loading and saving text files.

Edit

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

Preprocess

Where all the preprocessing of files is done before generating any output. These functions alter original text files only.

Generate

Used for the generation of output from preprocessed files. These functions output files based on work done with preprocessing tools.

Tools

AutoMap contains a number of Editors and Viewers for the files. These allow the user to view support files used in preprocessing.

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, Previous, 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 tine with the Undo command.

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.

Quick Launch Buttons

These buttons mirror the functions found in the Preprocess menu.

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

File Menu

Description

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

Select Input Directory

Place all your text files in an empty directory and use **Select Input Directory** to load them into AutoMap. All files in the directory will be loaded.

Import Text

Similar to **Select Input Directory** but AutoMap asks for the type of text encoding to use.

Let AutoMap Detect
AutoMap will attempt to import text with the best possible encoding method.

UTF-16
A variable-length character encoding for Unicode, capable of encoding the entire Unicode repertoire.

UTF-16LE (Endian)
Data types longer than a byte with the most significant byte (MSB) first.

UTF-16BE (Big Endian)

Data types longer than a byte with the most significant byte (MSB) last.

Windows-1252

a character encoding of the Latin alphabet, used by default in the legacy components of Microsoft Windows in English and some other Western languages. It is one version within the group of Windows code pages. The use of Unicode (often in UTF-8 form) is slowly replacing use of 8-bit "code pages" such as Windows-1252.

ISO-8859-1 (Western)

a standard character encoding of the Latin alphabet. It is less formally referred to as Latin-1. In June 2004, the ISO/IEC working group responsible for maintaining eight-bit coded character sets disbanded and ceased all maintenance of ISO 8859, including ISO 8859-1, in order to concentrate on the Universal Character Set and Unicode.

Save Preprocessed Text Files

Saves all text files at the highest level of preprocessing. This procedure can be done any number of times during processing. Just make sure if you want to keep a set of files to save them to an empty directory.

Exit AutoMap

Closes all files and exits AutoMap.

Edit Menu

Description

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

Set Font

Allows the user to change the font used in the display window.

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

Removes the last Preprocessing done to the text. Does only one step at a time. Multiple Undos can be performed on the text.

Remove Extra White Space

Removes all cases of multiple white spaces and replaces them with a single space.

Remove Punctuation

The Remove Punctuation function removes the following punctuation from the text: **.,:;' "()!?-**. The option is to remove completely or replace with a white space.

Remove Symbols

The list of symbols that are removed: **~`@#$%^&*_+={}[]\|/<>**. The option is to remove completely or replace with a white space.

Remove Numbers

Removing numbers will remove not only numbers as individual concepts but also removes numbers embedded within concepts. The option is to remove completely or replace with a white space.

Convert to Uppercase

Convert to Uppercase changes all text to either **UPPERCASE**.

Convert to Lowercase

Convert to Lowercase changes all text to either **lowercase**.

Apply Stemming

Stemming removes suffixes from words. 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. Two Stemmers are available, **K-Stem and Porter**.

Apply Delete List

A Delete List is a list of concepts to be removed from a text files. It is primarily used to reduce the number unnecessary concepts. By reducing the number of concepts being processed run times are decreased and semantic networks are easier to understand. This also helps in the creation of a semantic network in reducing the number of superficial nodes in ORA.

Apply Generalization Thesauri

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.

Generate Menu

Description

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

Concept List

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).

Semantic List

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

Parts of Speech

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.

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. Semantic Networks created can be displayed in ORA.

MetaNetwork DyNetML

Assigns MetaNetwork categories to the concepts in a file. This is used to create a DyNetML file used in ORA.

BiGrams

BiGrams are two adjacent concepts in the same sentence (two concepts can not cross sentence or paragraph boundary). 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.

Text Properties

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

Named Entities

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

Feature Extraction

The Feature Selection creates a list of concepts as a TF*IDF (Term Frequency by Inverse Document Frequency) descending order. This list can be used to determine the mot important concepts in a file.

Suggested MetaNetwork Thesauri

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.

A Meta-Network Thesaurus associates concepts with the following meta-network categories: **Agent, Knowledge, Resource, Task, Event, Organization, Location, Action, Role, Attribute, and a user-defined categories**.

Union Concept Lists

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.

Content Section

Content

This section contains general explanations of the functions of AutoMap. It details the "What it is" aspect.

Details pertaining to running AutoMap are contained in the other sections.

Anaphora

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.

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.

Example

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 justs 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.

Bi-Grams

Description

BiGrams are two adjacent concepts in the same sentence. The two concepts can not cross sentence or paragraph boundary. 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.

Threshold:

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 >= Global Threshold. For **Local Threshold** a Bi-gram is detected if the number of its occurrences in EACH text is >= 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
```

Then it qualifies for GlobalThreshold: 2+3+1 >= 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.

Text:

```
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.

The original files

```
John Doe is actively involved in several industry and civic
associations.
```

A Delete List containing three concepts:

```
is, in, and
```

A Generalization Thesaurus:

```
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

Description

Original file:

```
John Doe is a business leader. John Doe is a president of the
John Doe business.
```

Delete List :

```
is a of the
```

The final numbers are:

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

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:

Original file:

```
  John Doe works at John Doe Inc.
```

Concept List:

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

Delete List:

```
  at
```

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

```
  John, Doe, works, John, Doe, Inc
```

Generalization Thesaurus:

```
John_doe_inc
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.

Delete Lists

Description

A Delete List is a list of concepts to be removed from a text files. It is primarily used to reduce the number unnecessary concepts. By reducing the number of concepts being processed run times are decreased and semantic networks are easier to understand. This also helps in the creation of a semantic network in reducing the number of superficial nodes in ORA.

You can create Delete Lists for each set of files. This allows you to refine the final output better.

There are two types of adjacency: direct and rhetorical. The use of either one will be dictated by your need to maintain the original distance between concepts.

Points to Remember

The Delete List is **NOT** case sensitive. **He** and **he** are considered the same concept. Placing either one in the Delete List will move all instances.

You can create Delete Lists from a text editor or use the tools in AutoMap to assist in creating a specially-tailored Delete List.

All Delete Lists can be edited.

Multiple Delete Lists can be used on the same set of files.

Any Delete List can be saved and used for any other text files.

Adjacency

Direct Adjacency

This removes the concepts from the list totally. The concepts on either side then become adjacent to each other. This affects the spacing between concepts. If two concepts are deleted the two concepts surrounding them would then be adjacent.

Delete List:

in, the, of, he, on, a, it

Text:

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?

Direct Adjacency

Ted lives United States America. He lives dairy farm. He considers good life. Would he ever consider leaving?

In the original text is the sentence: **He lives on a dairy farm.** After the deletion the concepts on a are removed and the concepts **lives dairy** are now adjacent.

Rhetorical Adjacency

This removes the concepts but inserts a spacer **xxx** within the text to maintain the original distance between all concepts of the input file.

Delete List:

in, the, of, he, on, a, it

Text:

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?

Rhetorical Adjacency

```
Ted lives xxx xxx United States xxx America. He lives xxx xxx
dairy farm. He considers xxx xxx good life. Would he ever
consider leaving?
```

In his example the same two words, **on a**, are removed from the original text. But with rhetorical adjacency spacers are inserted into the text. These two spacers maintain the exact distance between concepts as the original text. The results shows that there are two concepts between **Lives** and **dairy** but the substitution removes the actual concept from the result.

Reasons NOT to use a Delete List

For the most part using a Delete List on a file is a good idea. It removes many concepts that are unnecessary as they do not affect the meaning of the major concepts. But in some style of documents the meaning of two bi-grams could be drastically affected by two seemingly useless words. Most Delete Lists contain the concepts the and a. These two definite articles usually do not change the meaning of the text. But in some instances the meaning could be very substantial.

In a Field Operations manual there is a definite difference between the terms **a response** and **the response**. It is subtle, but very important.

So before you use a Delete List make sure that the words being included are not going to change the meaning.

Text Encoding

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.

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.

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.

Directionality

Directionality can be either uni-directional or bi-directional.

- **Uni-Directional :** When coding a link, only 1st => 2nd concept should be noted.

- **Bi-Directional :** When coding a link, both 1st <= 2nd and 1st => 2nd concept shall be noted.

Feature Selection

Description

The Feature Selection creates a list of concepts as a TF*IDF (Term frequency by Inverse Document Frequency) descending order. this list can be used to determine the mot important concepts in a file.

Date Styles

AutoMap understands certain styles of dates as shown below.

With the **month day, year** AutoMap detects the full date unless the day contains the numerical suffix.

```
January 1, 2009 => January 1, 2009, date
January 2nd, 2009 => January 2, date (the year was dropped)
```

The older military style date (with the abbreviated month) of **day month year** were all detected as currency. The modern **day month year** (with fully spelled out month) is detected as a date but drops the day.

```
1 FEB 09 => 1 FEB, currency
2 FEB 2009 => 2 FEB, currency
03 FEB 09 => 03 FEB, currency
04 FEB 2009 => 04 FEB, currency
5 February 2009 => February 2009,date dropped the day
```

The completely numerical style of date is detected as a number.

```
090301 => no entry
20090302 => no entry
```

the first one went undetected but the last three were correctly spotted as dates.

```
2009/4/1 => no entry
2009/04/2 => 2009/04, date (the day was dropped)
2009/4/03 => 2009/4/03, date
2009/04/04 => 2009/04/04, date
```

All detected as dates though some dropped off the year.

```
1/5/2009 => 1/5/2009, date
02/5/2009 => 02/5/2009, date
3/05/2009 => 3/05, date (the year was dropped)
04/05/2009 => 04/05/2009,date
```

All three detected as dates though some dropped the year.

```
June 1d, 2009 => June 1, date (the year was dropped)
June 2nd, 2009 => June 2, date (the year was dropped)
```

Both detected as dates but both dropped the day.

```
1 July 2009 => July 2009, date (the day was dropped)
02 July 2009 => July 2009, date (the day was dropped)
```

File Formats

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.

Text Formats The only format AutoMap can read. Uses the .txt file extension.

Other text formats

- **ASCII :** (American Standard Code for Information Interchange) is the lowest common denominator. There are actually two ASCII codes. The original 128 character, 7-bit code and the expanded 256 character, 8-bit code.

- **CSV :**(Comma Separated Value) A file type that stores tabular data. The format dates back to the early days of business computing. For this reason, CSV files are common on all computer platforms.
- **EBCDIC :**(Extended Binary Coded Decimal Interchange Code) is an 8-bit character encoding used on IBM mainframe operating systems such as z/OS, OS/390, VM and VSE, as well as IBM minicomputer operating systems such as OS/400 and i5/OS.
- **HTML :**(Hypertext Markup Language) The predominant markup language used for web pages. It is a text format but uses a tagging system which would be interrupted as concepts by AutoMap.
- **ISO/IEC 8859 :** Standard for 8-bit character encodings for use by computers.
- **RTF :**(Rich Text Format) A proprietary document file format developed by DEC in 1987 for cross-platform document interchange. Most word processors are able to read and write RTF documents.
- **UTF-8 :**(Uniform Transformation Format) It is able to represent any character in the Unicode standard, yet the initial encoding of byte codes and character assignments for UTF-8 is backward compatible with ASCII. For these reasons, it is steadily becoming the preferred encoding for e-mail, web pages, and other places where characters are stored or streamed.
- **XML :**(Extensible Markup Language) A general purpose markup language that allows users to define their own tags.

Format Case

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.

Named Entities

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).

Networks

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:

Agent	Interactio	Knowledg	Assignment	Employment

	n Network Who knows who Structure	**e Network** Who knows what-Culture	**Network** Who is assigned to what-Jobs	**Network** Who works where-Demography
Knowledge		**Information Network** What informs what-Data	**Requirements Network** What is needed to do what-Needs	**Competency Network** What knowledge is where-Culture
Tasks			**Precedence Network** What needs to be done before what-Operations	**Industrial Network** What tasks are done where-Niche
Organizations				**Inter-organizational Network** Which organizations work with which-Alliances

Parts of Speech

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 MetaNetwork. 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

CC	Coordinating conjunction	**PRP$**	Possessive pronoun
CD	Cardinal number	**RB**	Adverb

DT	Determiner	**RBR**	Adverb, comparative
EX	Existential there	**RBS**	Adverb, superlative
FW	Foreign word	**RP**	Particle
IN	Preposition or subordinating conjunction	**SYM**	Symbol
JJ	Adjective	**TO**	to
JJR	Adjective, comparative	**UH**	Interjection
JJS	Adjective, superlative	**VB**	Verb, base form
LS	List item marker	**VBD**	Verb, past tense
MD	Modal	**VBG**	Verb, gerund or present participle
NN	Noun, singular or mass	**VBN**	Verb, past participle
NNS	Noun, plural	**VBP**	Verb, non-3rd person singular present
NNP	Proper noun, singular	**VBZ**	Verb, 3rd person singular present
NNPS	Proper noun, plural	**WDT**	Wh-determiner
PDT	Predeterminer	**WP**	Wh-pronoun
POS	Possessive ending	**WP$**	Possessive wh-pronoun
PRP	Personal pronoun	**WRB**	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
IRR	Irrelevant term	16	409,103
NOUN	Noun	2	217,309
VERB	Verb	6	166,259
ADJ	Adjective	3	81,243
AGENTLOC	Agent	1	62,020
ANA	Anaphora	1	47,303
SYM	Noise	8	36,232
NUM	Number	1	15,178
MODAL	Modal verb	1	14,115
POS	Genitive marker	1	5,247
ORG	Organization	1	1,958
FW	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.

Example

Text:

```
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 ./.
```

Process Sequencing

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.

```
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.

Delete List

```
the
of
```

Generalization Thesaurus

```
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.
```

Remove Numbers

Description

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.

Text:

```
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.

Text:

```
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.

Text:

```
C3PO was a robot in the movie Star Wars.
```

Text after RemoveNumber:

```
C PO was a robot in the movie Star Wars.
```

Remove Punctuation

Description

The Remove Punctuation function removes the following punctuation from the text: **.,:;' "()!?-**.

Example

Text:

```
"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

Description

The list of symbols that are removed:
~`@#$%^&*_+={}[]\|/<>.

Example

Text:

```
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  would
  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.
```

Remove White Spaces

Description

Find instances of multiple spaces and replaces them a single space. After running removing extra whitespace all instances of multiple spaces are reduced to a single space.

The practice of putting two spaces at the end of a sentence is a carryover from the days of typewriters with mono-spaced typefaces. Two spaces, it was believed, made it easier to see where one sentence ended and the next began. Most typeset text, both before and after the typewriter, used a single space. Today, with the prevalence of proportionally spaced fonts, some believe that the practice is no longer necessary and even detrimental to the appearance of text.

Example

Between each of the bigrams is an increasing number of spaces. Removing extra white spaces will remove all but one of the spaces.

Text:

```
one space. two  spaces. three   spaces. four
spaces.
```

After removing extra white spaces

```
one space. two spaces. three spaces. four spaces.
```

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

Semantic Networks

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

Input File:

```
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
the,office,1
```

Stemming

Description

Stemming is a process for removing the commoner 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 be 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

Text:

```
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 United States of America. He live on
a dairy farm. He consider it a good life. Would he
ever consider leaving?
```

Porter Stemming

The **Porter stemmer** uses the Porter Stemming algorithm. Additionally, it converts irregular verbs into the verb's infinitive.

Porter Example

Text:

```
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 United States of America. He live on
a dairy farm. He consider it a good life. Would he
ever consider leaving?
```

Languages for Porter Stemming

Each language works it's stems differently. It's important to use the correct language files when stemming else you will obtain 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/span> 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.

Text Properties

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.
Then he buys milk and cookies.
```

milkAndCookies.csv

```
Number of Characters,83
Number of Clauses,3
Number of Sentences,3
Number of Words,16
```

Thesauri, General

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:

Text:

```
My name is John Smith and I live in the USA.
```

Generalization Thesaurus

```
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

Text:

```
My name is John Smith and I live in the USA.
```

Generalization Thesaurus

```
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

Text:

```
My name is John Smith and I live in the USA.
```

Generalization Thesaurus

```
John Smith,John_Smith
USA,United_States
```

Text after Generalization Thesauri applied with ThesauriContentOnly:

```
John_Smith United_States.
```

Stop Characters

Talk about the Arabic names with apostrophes and hyphens and the problems if you remove symbols before hand. If removed you could change whether or not your thesauri will find certain concepts.

If you convert case before a thesauri you could cause your thesauri not to find your concepts.

Why the Order of thesauri entries is Important

The order of the entries in the thesauri is important. If an entry toward the beginning contains part of an entry that follows it then both substitutions will be done. This will result in an incorrect thesauri replacement. In the following example carter is substituted first causing incorrect substitutions later on.

Text:

```
O'Neill and Carter (Sam) are partners. Jacob Carter
is Sam's father.
```

Wrongly ordered Thesauri

```
Carter,Samantha_Carter
Jacob Carter,Jacob_Carter
Jacob,Jacob_Carter
```

Incorrect Result

```
O'Neill and Samantha_Carter (Sam) are partners.
Jacob_Carter Samantha_Carter is Sam's father.
```

When the order of the thesauri entries is corrected the resulting file now reads correctly.

Correct Thesauri

```
Jacob Carter, Jacob_Carter
Jacob,Jacob_Carter
carter,Samantha_Carter
```

Correct Results

```
O'Neill and Samantha_Carter (Sam) are partners.
Jacob_Carter is Sam's father.
```

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.

Thesauri, MetaNetwork

Description

Meta-Network 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.

There is a meta-network ontology which at the top level 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 whos are agents whether they be 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). It is up to the user's discretion what sub-category to place these agents in.

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. Anything that can be used for the completion of a job. (e.g. Use a car to drive from point A to point B or use money from a bank account to fund something).

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 (e.g.)

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 agents 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:

Original file:

```
Ted runs on a dairy farm. He milks the cows, runs
the office, and cleans the barn.
```

Delete List:

```
a, and, in, on, the
```

MetaNetwork Thesaurus:

```
Ted, agent
runs, task
dairy, resource
farm, location
He, agent
milks, task
cows, resource
office, location
cleans, task
barn, location
```

Concept List:

```
concept, frequency, relative_frequency, gram_type
He,1,1.0,single
Ted,1,1.0,single
barn,1,1.0,single
cleans,1,1.0,single
cows,1,1.0,single
dairy,1,1.0,single
farm,1,1.0,single
milks,1,1.0,single
office,1,1.0,single
runs,2,2.0,single
```

MetaList:

```
concept, frequency, relative_frequency, gram_type,
meta
He,1,1.0,single, agent
Ted,1,1.0,single, agent
barn,1,1.0,single, location
cleans,1,1.0,single, task
cows,1,1.0,single, resource
dairy,1,1.0,single, resource
farm,1,1.0,single, location
milks,1,1.0,single, task
office,1,1.0,single, location
runs,2,2.0,single, task
```

Thesaurus Content Only

Description

Using Thesaurus content only : After all concepts are replaced by key concepts from the thesaurus then only concepts matching those from the thesaurus will be kept.

Not using Thesaurus content only : Any concepts **NOT** in the thesaurus remain unaffected. All concepts, whether they are contained in the thesaurus or not, are output.

Thesaurus content only options:

Direct or rhetorical adjacency : means that original distances between concepts that represent the key concepts are neither visualized nor considered for analysis.

Rhetorical adjacency : means that the original distances between key concepts are retained and incorporated into later analyses. The original distances are visually symbolized by placeholders **(xxx)**.

Threshold, Global and Local

Description

The **Thresholds** refine the number of concepts to be included when saving 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 files can create large Concept List files.

```
Text 1 : See the boy named Dave. He has two balls.
One ball is red. and the other ball is blue.

Text 2 : On Monday Dave plays with the blue ball.
It's his favorite ball.

Text 3 : On all other days Dave plays with the red
ball.
```

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
  ball,5,1.0,0.12195121951219512
  balls,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
  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. What's needed is some pre-processing on the raw text using the Delete List, Stemming, and Thresholds

Removing contractions

Notice there's the contraction **it's**. In other texts there will probably be many more. To expand contractions before continuing run create a thesauri to expand all contractions. **it's** expands to **it is** as will any other contractions found in the thesauri file.

Removing plurals

Next we want to combine the concepts of **ball** and **balls**. They're both talking about the same item and we'd like to count them as one. Run **Stemming**. using KSTEM.

Running a Delete List

Using the Concept List Viewer 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
ball,6,1.0,0.18181818181818182
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,2,0.3333333333333333,0.06060606060606061
play,2,0.3333333333333333,0.06060606060606061
red,2,0.3333333333333333,0.06060606060606061
see,1,0.16666666666666666,0.030303030303030304
two,1,0.16666666666666666,0.030303030303030304
with,2,0.3333333333333333,0.06060606060606061
Total,33
Mean,1.65
StDev,0.0
```

There's a definite difference between the two list. Originally there were 25 individual concepts. Now there's a total of 20. With the thresholds we will reduce this even farther.

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 to be included in the Union Concept List a concept must appear a total of two or more times in **all** text files.

Create a new concept List.

```
Words,Frequency,Relative Frequency,Relative
Percentage
ball,6,1.0,0.2857142857142857
be,2,0.3333333333333333,0.09523809523809523
blue,2,0.3333333333333333,0.09523809523809523
dave,3,0.5,0.14285714285714285
other,2,0.3333333333333333,0.09523809523809523
play,2,0.3333333333333333,0.09523809523809523
red,2,0.3333333333333333,0.09523809523809523
with,2,0.3333333333333333,0.09523809523809523
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 (ball and dave) 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
ball,3,1.0,0.6
be,2,0.6666666666666666,0.4
Total,5
Mean,2.5
StDev,0.0
```

Union Concept List

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, and after review, can be determined as concepts that can be added to the Delete List.

- 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 : Takes the frequency of any concept divided by the highest value of any frequency

Relative Percentage : Add all the relative_frequency values then divide a concept’s relative_frequency by that value.

Example

Start with two (or more) texts.

fireman.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 for all texts loaded.

Using in Excel

This file can then be sorted in Excel. Open the file in Excel. All the data will appear in a single column. To separate it select the column with the data. Then select from the menu **Data => Text to Columns**. 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.

Window Size

Description

Changing the window size determines the span which connections will be made. The larger the window size, the more connections within that window. The window slides allow the text changing by one concept as each window is finished.

Starting at the beginning it will analyze the set of concepts. The window will then move one concept to the right and create a new window to analyze. This will continue until it reaches the end of the text.

Example

Text:

```
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. Too small and important links may be missed. Too large and too many concepts are connected and important links may be overwhelmed.

```
Dave likes milk and cookies but John likes
cauliflower
```

The above sentence contains nine concepts. Manually reviewing this sentence you can see 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** then cauliflower would also be associated with Dave.

Tools

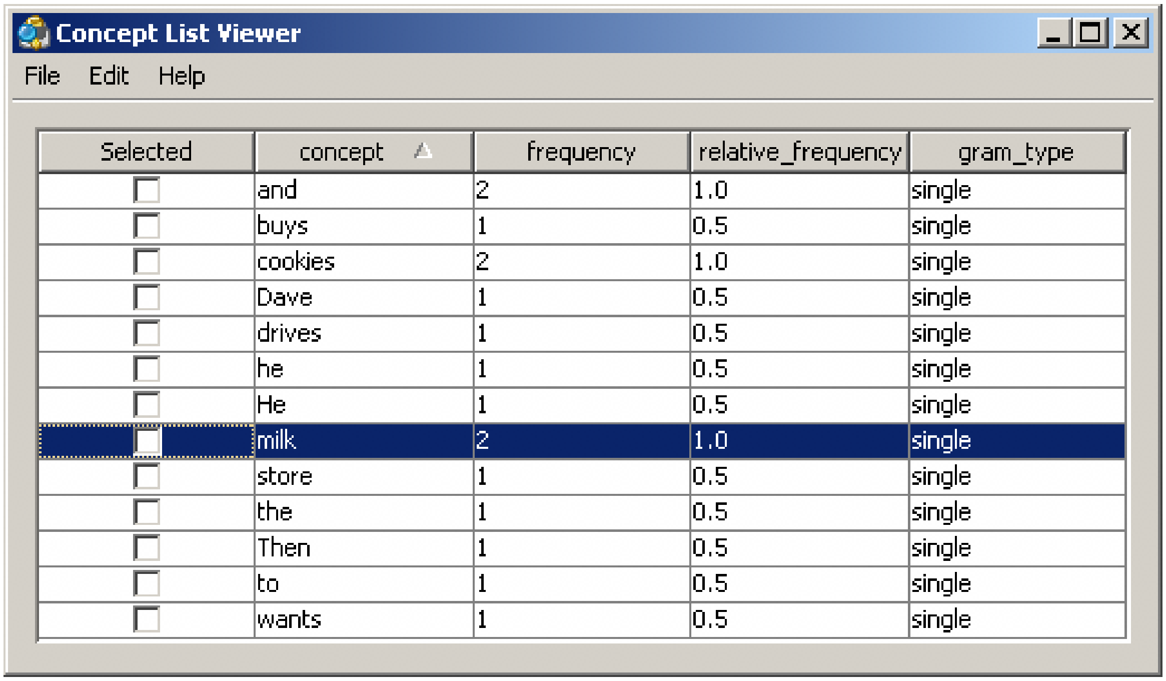
This section contains descriptions of the tools contained in AutoMap. They include:

1. **Concept List Viewer**
2. **Delete List Editor**
3. **Semantic List Viewer**

Concept List Viewer

Description

The **Concept List Viewer** is used to view and edit any concept list 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**.



Selected	concept ▲	frequency	relative_frequency	gram_type
□	and	2	1.0	single
□	buys	1	0.5	single
□	cookies	2	1.0	single
□	Dave	1	0.5	single
□	drives	1	0.5	single
□	he	1	0.5	single
□	He	1	0.5	single
□	milk	2	1.0	single
□	store	1	0.5	single
□	the	1	0.5	single
□	Then	1	0.5	single
□	to	1	0.5	single
□	wants	1	0.5	single

From the Pull Down Menu select **`Tools => Concept List Viewer`**.

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.

Selecting Concepts

Under the **`Edit menu`** the viewer gives you options for selecting/deselecting multiple concepts. The user can also manually place and remove individual check marks by clicking in the box.

Select All : Places a check mark in every check box.

Select None : Clears all check marks from the concept list.

Select Greater Than : Allows the user to set the frequency threshold which will select only those concepts whose frequency is **That value or greater**.

You can also select individual concepts by placing a check mark in the box next to the concept's name.

Compare Files

Two concept files can be compared to one another to find which concepts occur in both files. This option will also display (by virtue of red and green stripes) which concepts occur in only one file.)

milkAndCookies.txt

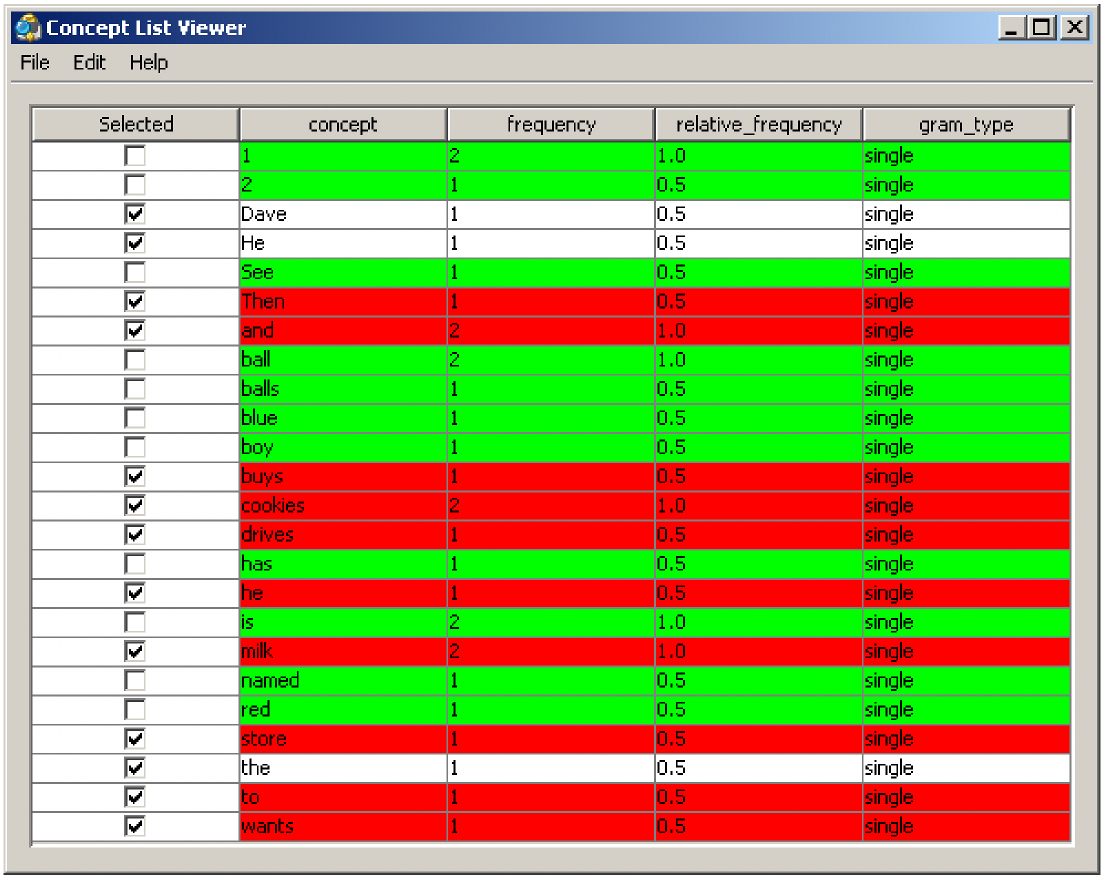
```
Dave wants milk and cookies. He drives to the store.
Then he buys milk and cookies.
```

theBoy.txt

```
See the boy named Dave. He has 2 balls. 1 ball is
red. 1 ball is blue.
```

From the viewer menu select **`File => Open File`** and navigate to the first file in the set to use. From the viewer menu select **`File => Compare Files`** and navigate to the second file in the set.

For demonstration purpose I have done a **`Edit => Select All`** to display which concepts belong to the first file before using **`File => Compare Files`**.



Selected	concept	frequency	relative_frequency	gram_type
☐	🟩 1	2	1.0	single
☐	🟩 2	1	0.5	single
☑	Dave	1	0.5	single
☑	He	1	0.5	single
☐	🟩 See	1	0.5	single
☑	🟥 Then	1	0.5	single
☑	🟥 and	2	1.0	single
☐	🟩 ball	2	1.0	single
☐	🟩 balls	1	0.5	single
☐	🟩 blue	1	0.5	single
☐	🟩 boy	1	0.5	single
☑	🟥 buys	1	0.5	single
☑	🟥 cookies	2	1.0	single
☑	🟥 drives	1	0.5	single
☐	🟩 has	1	0.5	single
☑	🟥 he	1	0.5	single
☐	🟩 is	2	1.0	single
☑	🟥 milk	2	1.0	single
☐	🟩 named	1	0.5	single
☐	🟩 red	1	0.5	single
☑	🟥 store	1	0.5	single
☑	the	1	0.5	single
☑	🟥 to	1	0.5	single
☑	🟥 wants	1	0.5	single

Unhighlighted with a check mark : These concepts appear in both files.

Red highlight with a check mark : These concepts appear only in milkAndCookies.txt.

Green highlight without a check mark : These concepts appear only theBoy.txt.

The importance of this function allows you to see which concepts are unique to individual files.

Create a Delete List

To create a Delete List place a check mark in the **Selected** column of all the concepts to include. Then from the Concept List Viewer menu select `File => Save as Delete List`.

Navigate to the folder you want to save your delete list and give it a file name. AutoMap will not save the file unless you give it a file name with an extension.

NOTE : The file type can be either **.txt** or **.csv**.

This new file can now be loaded and applied as a new Delete List.

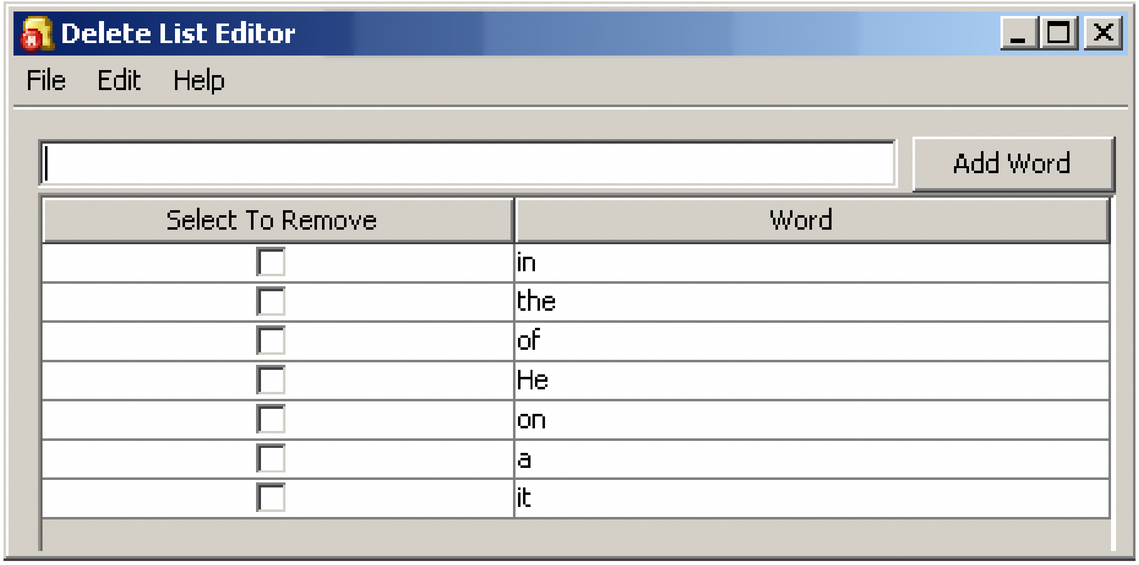
Delete List Editor

Description

With the **Delete List Editor** you can modify an existing Delete List by adding and subtracting concepts. The new Delete List can be saved under a new name from the previous Delete List.

Procedure

From the Pull Down Menu select `Tools => Delete List Editor`.



Select To Remove	Word
☐	in
☐	the
☐	of
☐	He
☐	on
☐	a
☐	it

From this window you can:

- **Add New Concepts:** In the textbox above the list type in a new concept. Then click **[Add Word]**. Your new concept will be added to the list.
- **Removing Existing Concepts :** Click in the check box next to the concept to remove. The next time you save the Delete List it will be saved without the checked concepts.

  NOTE : No concepts are added or deleted until you actually save the file.

- **Create New Delete List :** From the viewer Pull Down Menu select `File => Save as Delete List`. AutoMap will prompt you to select a directory and give the file a new file name.

Note : Make sure to give the file the **csv** extension.

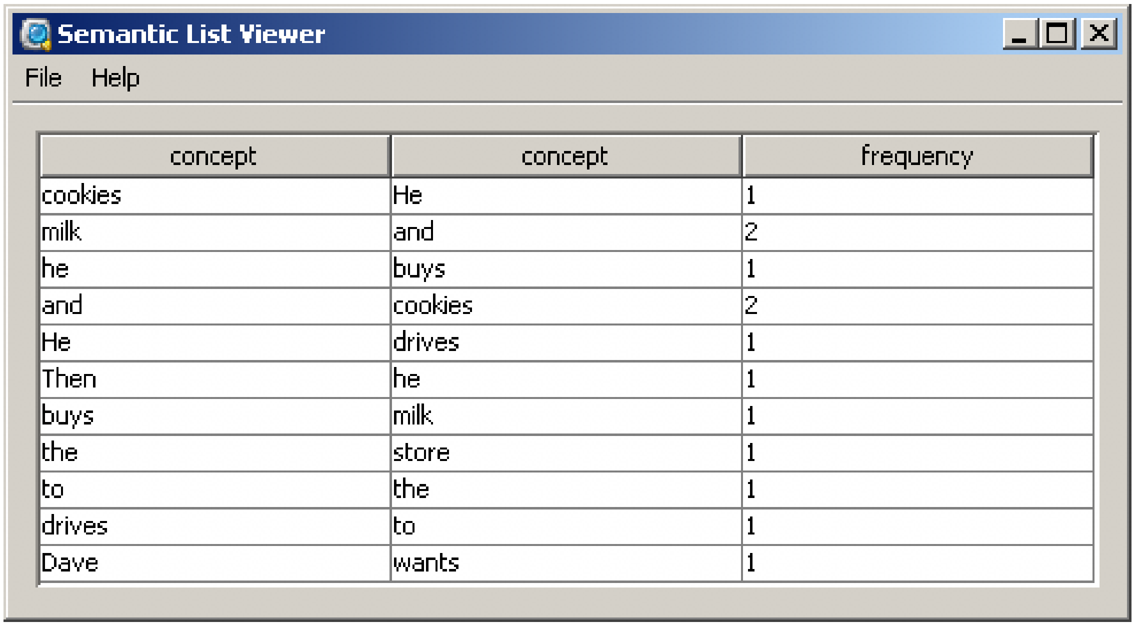
Semantic List Viewer

Description

With the **Semantic List Viewer** you can view an existing Semantic List created by AutoMap.

Procedure

From the Pull Down Menu select `Tools => Semantic List Viewer` From the viewers Pull Down menu select `File => Open File`. Navigate to a semantic list and click **Open**.



concept	concept	frequency
cookies	He	1
milk	and	2
he	buys	1
and	cookies	2
He	drives	1
Then	he	1
buys	milk	1
the	store	1
to	the	1
drives	to	1
Dave	wants	1

The viewer displays all combinations of concepts in a file and their frequency depending on the parameters given to create the list.

- **Select Directionality :** Directionality can be either **Unidirectional** : making connections between two concepts by only looking forward in the text limited by the window size OR **Bidirectional** : making connections between two concepts by looking both forward backward in the text limited by the window size.
- **Select Window Size :** The distant concepts can be and still have a relationship to one another. Only concepts in same window can form statements
- **Select Stop Unit :** The limiting factor for concepts to make a connection. Defined as Word, Clause, Sentance, or Paragraph.
- **Select Number of Sentences :** The number of sentances that can be included with the Window Size parameter.

Script

Description

All of AutoMap's functions are readily found in the Script file.

Two items are necessary when using the script.

1. Knowledge of the Command Run Window.
2. Understanding of XML formatting.

AM3Script

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 choosen 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 />

`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>
</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 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 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.t
  xt" />
</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>
```

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.