Workshop in Applied Analysis Software
MY591

Introduction to Wordstat and QDA Miner

Instructor:

Prof. Ahmet K. Suerdem (Istanbul Bilgi University and London School of Economics)
Contact: A.K.Suerdem@lse.ac.uk

Course Convenor (MY591)
Dr. Aude Bicquelet (LSE, Methodology Department)
**WordStat and QDA Miner are** computer assisted text analysis tools designed by Norman Peledeau and distributed by Provalis Research ([http://www.provalisresearch.com/index.html](http://www.provalisresearch.com/index.html)). More specifically, **WordStat** can be used as a text mining tool for extraction of themes and trends or achieve measurement with a quantitative content analysis method. **WordStat** is integrated with **Simstat**, a statistical data analysis tool and/or **QDA Miner**, a qualitative data analysis software. QDA Miner is a mixed-model qualitative data analysis software package for coding, annotating, retrieving and analyzing small and large collections of documents and images and with some text mining tools.

You can download the evaluation copies and manuals for both software for free (for 1 month). You can also purchase the bundle for 945$ from Provalis Research.

This handout is based on the following references: (Peledeau 2004); (Peledeau 2005) and various sources on text analysis and content analysis. For content analysis (Weber 1990; Neuendorf 2002; Krippendorff 2004) are the classics. For text analysis and qualitative data analysis please refer to your discipline’s classics.

**Computer Assisted Text Analysis: A basic introduction**

Text analysis (TA) is a tool for social analysis which includes a wide range of techniques for organising texts, for coding meaning units, for developing coding frames (categorization schemes), for describing and coding the relationships between different meaning units and generally for revealing patterns between these units. Usually, the terms text analysis and content analysis are used interchangeably although they point to different methodologies. Text analysis is a general field of practice employed by many disciplines from literary criticism to artificial intelligence where the purposes and methods differ accordingly. On the other hand, content analysis (CA) is a method for text analysis and provides a theory and set of techniques for extracting information from textual data regardless of the discipline. Different from TA, CA aims to quantify and categorise the content or meaning of certain textual configurations (words, word combinations, sentences, etc.). The difference lies in the interpretation framework for analysis. Text analysis is also concerned with extracting and categorisation of meaning; however, this concern is more determined according to the discipline and purpose of the research in question. Text analysis may converge to content analysis as the coding frame becomes more structured (sometimes called qualitative content analysis) and as the aim of analysis becomes hypothesis testing or diverge from it as the methods of analysis become more interpretive (there are a variety of interpretive text analysis methods from hermeneutics to semiotics).

In this context, both WordStat and QDA Miner are text analysis tools which can also be used for content analysis. For content analysis, WordStat is more convenient for quantitative (automatic coding according to a structured dictionary) while QDA miner for qualitative content analysis (interpretive, thematic coding according to a coding frame). For text analysis on the other hand, WordStat detects patterns according to word co-occurrences, QDA Miner does the same for in-vivo coded code co-occurrences. In terms of their underlying methodological functions these two software share a common basis; they differ in the sense
that where the former focuses on the automatic coding while the latter on human coding. They can both be used for content analysis; text analysis and qualitative data analysis.

QDA Miner may essentially be used as a tool for qualitative data analysis (WordStat on the other hand is limited to the analysis of computer readable tokens). Text analysis is not limited to the analysis of linguistic expressions in written form. Krippendorf (2004) offers a more general notion of text: “the term may refer to a variety of data the texture of which becomes apparent through its identifiable and distinguishable components” p. 8. In this respect, any qualitatively collected data can become the subject of text analysis. Thus, QDA Miner may be used to analyze interview or focus-group transcripts, legal documents, journal articles, books, as well as photographs, paintings, and other types of visual documents. These data need not to be converted to texts, QDA Miner allows to code data even in the form of images. A word of caution though: QDA Miner is a tool and text analysis does not mean to replace qualitative analysis. Particular context of situation analyzed and the domain of the text is essential for qualitative analysis and there is variety of methods putting the burden on the interpretation process. Making sense of computer-assisted text analysis is not intrinsic to text itself; interpretation has to be provided by means of more hermeneutic methods. Detecting patterns from just within the text itself is a sort of “Kabbalistic” practice; it is like seeing patterns everywhere. The definition of the categorization scheme or the coding process itself must be grounded into the social context of the study by means of the established interpretive practices of the discipline (such as discourse analysis, semiotics, ethnography etc…). Moreover, interpretation is not theory free; how meaning is defined is dependent on one’s theoretical frame of reference. Thus, computer assisted text analysis tools are by no means meant to replace the hermeneutic circle which involves an interactive interpretation process taking into account the social context, text itself and the theoretical framework of the interpreter (Kelle 1995; Kelle 1997).

Rather than just containing an information about linguistic expressions, textual data is a rich source containing information about social structure, norms, values, attitudes and political positions about the social context where the text was produced. Depending on the purpose, the textual data may belong to a single semantic domain, e.g. political speeches, or may cut across a number of semantic domains. Furthermore, they may consist of one particular text type, for example only personal letters or articles or speeches, etc., or they may include texts belonging to different text types. The relevance of texts for the purpose is determined during the research design, sampling and data collection phases, which proceed the analysis phase. There are a variety of techniques for this purpose which are beyond the purposes of this workshop (reader may refer to (Bauer and Gaskell 2000) for a comprehensive guide concerning these techniques). For this workshop, the competence in these techniques is taken for granted. Thus, when using computer assisted text analysis software always consider that they are tools to assist you to describe, classify, interpret or make inferences about a social context based on a corpus of textual data relevant to particular context(s) of situation to be investigated. A tool becomes a fetish if not complemented by the theories and methodologies of your specific discipline.
Introduction to QDA Miner

QDA Miner is a qualitative data and text analysis software package for coding textual data and annotating, retrieving and reviewing coded data and documents. Besides its text analysis features, QDA Miner also provides a wide range of exploratory tools to identify patterns in codings and relationships between assigned codes and other numerical or categorical variables.

Some basic concepts

- **Project**: QDA Miner keeps all documents, coding schemes, codes, and notes in a set of files called a “project”. QDA Miner projects consist of multiple cases.
- **Case**: is the basic unit of analysis of a project. It typically represents an individual, an organization, or a group.
- **Variable**: A case can contain several documents as well as numerous alphanumeric, numeric, categorical, date or Boolean variables operationalising the attributes of cases. One of the unique features of QDA Miner is its ability to explore relationships between any one of these variables and codes manually assigned to documents. For example, you can assess how the content of an interview is related to the interviewee’s gender or age, or how it relates to specific answers to a closed ended question.
- **Code**: Coding is the process of marking text segments (or parts of images or sections of a video recording) as meaning units. The segments interpreted as connoting a similar meaning are marked with a label, the code, that is usually associated with a description of what the code means, common keywords for the text segments and, perhaps, a general interpretation of them.
The Working Environment

The large DOCUMENT window on the middle side of the screen is the main working space. This window is used to view and edit documents and images and assign codes to text segments or image areas. When a case contains more than one document or image, you can switch between them by selecting the document/image name from the list box located in the top left corner of this window. To the right of this window is a gutter area. This area is used to display codes assigned to text segments and image areas. A second gutter area is displayed at the bottom of images. Colour brackets are used to indicate the physical limits of the coded segment. This area is used to review coding, remove or change assigned codes, and attach notes to any coded segments.

Step by step guide for working with QDA Miner: How to get to and work with the working environment:

Opening a Project from the Introductory Screen

When you start the program the following window appears:
The last two options are quite straightforward. The first option, create a new project needs some explanation:

- **Four ways four Creating a New Project:**

  1. **Creating a project from a list of documents/images**

    Click on the folder where you keep your text corpus and select the documents (compatible files: MS Word, WordPerfect, RTF, PDF, plain text files or HTML as well as image files such as BMP, JPG, GIF, PNG and WMF). Click on the Add button at the bottom. Once all files have been selected, click the Create button. You will be asked to specify the name of the project that you want to create.

  2. **Creating a Blank Project**

    After naming and describing your variable, click on the Data type: A variable may contain a document or an image to be manually coded, but can also consist of a numeric value, a date, a Boolean value (true or false), etc. Select your data type.
If you select a nominal/ordinal data type; define the values by *clicking the Edit button*. The following window will appear:

![Variable definition window](image)

You can start typing values (one value per line) in the large edit box. If the current variable uses the same values as another existing variable, you may also establish a link to this other variable so that they will share the same list of values. *Click OK*. Return to Variable definitions window.

You don’t need to do that with other variable types. After defining each of your variable don’t forget to *click the Add` button to add the defined variable to the list. You can edit or remove` a defined variable.

When you have finished defining the structure of the new project, *click the OK` button. You will be asked to specify the name of the project that you want to create.

3. **Importing an Existing Data File**

QDA Miner allows you to directly import data files from spreadsheet and database applications, as well as from plain ASCII data files (comma or tab delimited text). You may refer to the Manual (p. 17) and *Formatting spreadsheet data* (p. 43 this handout) section on for further information about the Formatting requirements, supported features and limitations.

4. **Importing Existing Documents as a New Project: Document Conversion Wizard**

The Document Conversion Wizard is a utility program used to import one or more documents into a new project file. This tool supports the importation of numerous file formats including plain ASCII, Rich Text Format, MS Word, HTML, Acrobat PDF files, and WordPerfect documents. It may be instructed to split large files into several cases and to extract numeric and alphanumeric data from these files. The Document Conversion Wizard may be run either as a standalone application or from within QDA Miner. When no splitting, transformation or extraction of information from documents are necessary, an easier method would be to use Creating a Project from a List of Documents.

After creating your project; following operations may be useful for fine tuning and organising your project.
Project Menu

Project Description and Notes
QDA Miner allows you to specify a description for the project as well as write general notes to yourself or to be shared with other people working on this project.

Project>Notes

Enter your comments or observations. This can be considered as a memo.

Security & Multi-users Settings
It is possible to limit the number of people who can access and edit a project or limit the type of operations that can be performed by certain individuals by creating user accounts and requiring people to provide a user name and password when they access the project. However, the multi-user account feature is especially useful and necessary for assessing the inter-coder reliability (see Inter-coder reliability).

When setting a project to support multiple users, one of these users should be able to control access to the project, create and delete user accounts and define passwords. This user is commonly known as the Admin(istrator). Both the default login user name and password is ADMIN. You may change this for security.

Project>Teamwork > User settings

By default, opening a project without using the user log screen gives the user all administrator access rights. Enabling the Users must log option will display a logon dialog box prompting the user to enter a valid user name and password in order to access the project file.

• To add a new user account:
• **To define the user access rights:**
  Select the user for which you want to define or edit access rights. In the list of available features to the right of the dialog box, select the features you want this user to have access to and clear the features that you do not want him to access. For example, when creating several user accounts to establish intercoders agreement, you can deactivate the **View other users’ coding** option.

Creating a duplicate copy of a project
The TEAMWORK | DUPLICATE PROJECT command from the PROJECT menu provides an easy way to store an exact copy of the existing project under a different project name or into a different location. To create a new project with an identical structure as the current project but with only a subset of cases, use the EXPORT | PROJECT FILE command instead. To create such a file, set the case-filtering conditions of the active project to display the cases as they should be stored in the new data file (see Filtering Cases) before running the exportation command (see Exporting Selected Cases).

Sending a Project by E-mail
The TEAMWORK | SEND BY EMAIL command provides an easy way to electronically send a project to another team member or to any other interested party. The custom project is then stored in a compact ZIP archive and sent to the user as an email attachment. But before doing this you must do the necessary arrangements with the **Outlook**. Otherwise you may get an error message.

Exporting Coded Segments
In some situations the format in which the data has been collected does not correspond to the design of a research project or is not appropriate for the kinds of analyses that need to be performed. For example, if one collects transcripts of group discussions and would like the unit of analysis to be the individuals, one will need to split the transcript so that interventions by each speaker will be stored in separate cases, creating for each participant a single new document of everything this person said. Another example would be when one has several structured documents but is only interested in analyzing specific sections of them and would like to eliminate unnecessary sections. Alternatively, one may want to regroup, under a single document, segments about a specific topic currently found in several document variables. The **Export Coded Segments feature** is a flexible data restructuring tool that may be used to perform all the above changes. It may also be used to transform a set of unstructured individual interviews into a structured database, storing answers to different questions into separate document variables or regrouping text about a
specific topic discussed by an interviewer at different moments under a single
document variable.
In the following example, the documents (METHO; GROUPNO; DATEGRGROUP)
were coded as John, Steven, Lida etc…

EXPORT | CODED SEGMENTS command from the PROJECT

- **SEARCH IN** - This option allows you to specify which document variables the
  extraction should be performed in where the project has more than one document
  variables (i.e. focus group transcripts; interview transcripts; field observations)
- **CODES** - This option allows you to select the codes defining the segments that
  should be extracted.
- **CASE SPLITTING** - This option is used to specify whether the original case
  structure should be preserved or whether existing cases should be split into several
  ones based on the coding made of their documents. If the **Every coded segments**
  option is selected, the program will extract each coded segment associated with
  any one of the specified codes and will store the segment in a separate case. For
  example, if you have coded parts of a focus group transcript as three times John,
  two times Steven, and one time Lida etc.. then selecting this option will produce
  altogether 6 cases, one for each code. If **Merged coded segments** is chosen, a
  new case will be created for each selected code. This new case will contain
  all segments associated with this code. If we carry on with the above example, this
  time we’ll have only John, Steven, and Lida etc.. as (3) cases.
- **VARIABLE SPLITTING** - This option can be used to specify whether extracted
  segments should be stored in a single document variable or in different ones.
- **ADD VARIABLES** - This drop-down checklist box may be used to append the
  values of existing variables to the exported data file along with the extracted text
  segments.

**Exporting Documents**
The EXPORT | DOCUMENTS command saves all documents in a project file
associated with one or several document variables into separate files such as text files.
After creating your project you can work through it by clicking on the commands on the Menu:

**Cases Menu**

**Adding and Deleting Cases**

*Cases > Add*

To enter a new value, click the date entry cell located to the right of the variable name that you want to edit. If the variable is numeric or alphanumeric, you can start typing the data you want to store in this variable. For categorical variables, dates and Boolean values, press the F2 key or double-click the cell to display the list of available values or open a date editor. The tab key will take you to the next variable. To import a graphic file into an image variable, double-click the data entry cell or press F2. An open dialog box will appear, allowing you to select the graphic file to import. To enter text or import an existing file into a document variable, double-click the data entry cell or press F2. A text editor window will appear. You can start entering the text you want to store in this document variable.

You can append New Documents and Images: *Cases > append new cases/images*

**Filtering Cases**

The FILTER command in the CASES menu temporarily selects cases according to some logical condition. You can use this command to restrict analysis to a subsample of cases or to temporarily exclude some subjects. The filtering condition may consist of a simple expression, or include up to four expressions joined by logical operators (i.e., AND, OR). You can append New Documents and Images: *Cases > append new cases/images*
To retrieve a previously saved filter, click the button and select from the displayed list the name of the filter you would like to retrieve.

Exporting Selected Cases
You may want to use your data in a different format for alternative analyses. For example, you may want to export some of your variables to make some statistical analyses with SPSS: The PROJECT | EXPORT command saves a copy of the current project under a different name or exports the project to another file format like Excel, CSV, XML etc...

Setting the Cases Descriptor and Grouping
By default, cases are listed in their order of creation and are identified by their physical position in the data file (e.g. case #1, case #2, etc.). However, you can edit the list order, group cases in categories, and define a custom descriptor that will be used to identify a case based on the values contained in one or more variables. For example, if you want to group presidential candidate speeches into each candidate:

GROUPING/DESCRIPTOR command from the CASES
If you choose CANDIDATE the cases will be grouped by each candidate’s name; if you choose TOPIC then the cases will be grouped to topic: you can change case groupings according to your unit of analysis.

Appending a Document or Image into a Case
Select the case in which you would like to store the document: Cases>Append Documents/Images
Select the document(s) you want and Add. When you finish adding the documents Append. Click OK on Variable Selection.

You’ll see the documents in a separate case named Missing. Go to the Variables window and define the values for the documents for each variable. For our example, set the CANDIDATE to Bradley and TOPIC to Announcement. Also define the values for other variables if necessary. The new document automatically will be grouped in the case it belongs to:

Variables Menu
Nothing much to add as we have already discussed how to create new variables, set the values for each variable for the cases etc… in Creating a Blank Project section. The Menu is quite intuitive and the reader may refer to the Manual for the details if needed.

Codes Menu
Coding and indexing is necessary for systematic qualitative data or text analysis. Coding allows to identify variables and values, therefore allows for systematic analysis of data (reliability i.e., codes are assigned to similar meaning units and ensures enhanced construct validity, i.e. that you look at things allowing to measure your concepts).

- **Don’t start coding before you have good idea on your coding strategy!**
  - either your code book is determined by your research questions and associated theories, frameworks, analysis grids
  - or you really learn how to use an inductive strategy like "grounded theory".
QDA Miner allows for both strategies: You can either follow a top down approach and start with a structured codebook determined according to a framework or you can follow a bottom up approach: first create individual codes and structure them in a grounded way.

You can find more details on how to construct a codebook in The Content Analysis Guidebook by Kimberly A. Neuendorf among other clear instructions about content analysis. You can find some codebook examples at Neuendorf’s link: http://academic.csuohio.edu/kneuendorf/content/hcoding/hcindex.html

For more on interpretive coding please refer to the qualitative data analysis sources (For example: (Glaser and Holton 2007; Corbin and Strauss 2008))

- **Example: Party Manifesto Project Codebook**
  (Excerpts for a brief demonstration: For more details please refer to http://manifestoproject.wzb.eu/documentation/Handbookv4.pdf )

  - **Quantification: The Coding Unit**
    The coding unit in any given program is the ‘quasi-sentence,’ defined as an argument. An argument is the verbal expression of one political idea or issue. In its simplest form, a sentence is the basic unit of meaning:

    We want worldwide peace.  
    We will reduce our military forces.

    More complex sentences should be decomposed into ‘quasi-sentences’ if the argument changes within the sentence:

    In international policy we shall take new initiatives. We will:
    - promote peace;
    - ban chemical weapons;
    - begin discussions for the removal of nuclear bases;
    - increase aid to developing countries;
    - take action to protect the status of refugees.
**Decision Rule No 1: Identifying Quasi-Sentences**

1. Copy the respective party program into the left column of a table with 2 columns, leave the right column for the codings).

- **Classification: The Standard Coding Frame**

  Table 1: 56 Standard Policy Preferences in Seven Policy Domains

<table>
<thead>
<tr>
<th>Domain 1: External Relations</th>
<th>Domain 2: Freedom and Democracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>101 Foreign Special Relationships: Positive</td>
<td>201 Freedom and Human Rights: Positive</td>
</tr>
<tr>
<td>102 Foreign Special Relationships: Negative</td>
<td>202 Democracy: Positive</td>
</tr>
<tr>
<td>103 Anti-Imperialism: Positive</td>
<td>203 Constitutionalism: Positive</td>
</tr>
<tr>
<td>104 Military: Positive</td>
<td>204 Constitutionalism: Negative</td>
</tr>
<tr>
<td>105 Military: Negative</td>
<td>Domain 3: Political System</td>
</tr>
<tr>
<td>106 Peace: Positive</td>
<td></td>
</tr>
<tr>
<td>107 Internationalism: Positive</td>
<td></td>
</tr>
<tr>
<td>108 European Integration: Positive</td>
<td></td>
</tr>
<tr>
<td>109 Internationalism: Negative</td>
<td></td>
</tr>
<tr>
<td>110 European Integration: Negative</td>
<td></td>
</tr>
<tr>
<td>Domain 5: Welfare and Quality of Life</td>
<td></td>
</tr>
<tr>
<td>403 Keynesian Demand Management: Positive</td>
<td>501 Environmental Protection: Positive</td>
</tr>
<tr>
<td>410 Economic Growth</td>
<td>502 Culture: Positive</td>
</tr>
<tr>
<td>411 Technology and Infrastructure: Positive</td>
<td>503 Social Justice: Positive</td>
</tr>
<tr>
<td>412 Controlled Economy: Positive</td>
<td>504 Welfare State Expansion</td>
</tr>
<tr>
<td>413 Nationalisation: Positive</td>
<td>505 Welfare State Limitation</td>
</tr>
<tr>
<td>414 Economic Orthodoxy: Positive</td>
<td>506 Education Expansion</td>
</tr>
<tr>
<td>415 Marxist Analysis: Positive</td>
<td>507 Education Limitation</td>
</tr>
<tr>
<td>416 Anti-Growth Economy: Positive</td>
<td></td>
</tr>
</tbody>
</table>

Besides these descriptions a codebook also includes sections such as the explanation of the project and coding exercise. Please refer to the Party Manifesto Project for more details.

- **Managing the Codebook in QDA Miner**

  The main function of QDA Miner is to assign codes to selected text segments and then analyze these codes. Available codes associated with a project are located in the CODES window in the lower left corner of the application workspace. Codes are grouped into categories. The list of available codes, known as the codebook, is displayed as a tree structure, in which categories are the nodes under which associated codes can be found.

  ![Codebook Tree Structure]

You can add, edit, delete, move, split merge and print codes.

**Adding a Code**

Codes > Add
The **Under** list box allows you to select the category under which this code will be stored. This control can be used both as an edit box to create a new category and as a list box from which you can select an existing category. For example, Topics is a category, Social is a subcategory. You can create categories to organise your codes. If you want to add the new code to an existing category, select the category name from the list of available categories and enter the new **Code name**. For example, enter Globalization as a code name under Topics/Economic category.

The **Description** option allows you to enter a definition or a detailed description of the code. You can use this section to specify coding instructions for the coders, along with examples or non-examples, and related codes that may be used in place of or in conjunction with this code.

The **Keywords** section allows you to enter words, word patterns, or phrases that are characteristic of the code. QDA Miner may then search text for any of those items using the KEYWORD RETRIEVAL command and retrieve sentences, paragraphs or documents containing any of those items. The following rules should be followed when entering keywords:
- Each item must be entered on a separate line.
- The underline character is used to join words of a phrase. Since space characters are not allowed in keyword entries, they are automatically transformed into underline characters.
- Wildcards such as * and ? are supported.

**Importing a Codebook**

**Codes>Import**

Quite often codebooks developed for a specific project may be used again for other coding purposes. QDA Miner can import a codebook and merge its codes and categories into the existing codebook. When merging codebooks, codes or categories whose names already exist in the current codebook are simply ignored.

WordStat categorization dictionaries may also be imported and transformed into the codebook. When importing a content analysis dictionary, low-level categories are automatically transformed into codes and their items are imported as keywords. Higher-level categories containing other content categories are transformed into new categories. For example, the importation of the following WordStat dictionary:
will create two codes, NORTH-AMERICA and SOUTH-AMERICA, under the ORIGIN category. Items stored under the two codes will automatically be imported as keywords.

Coding Documents and Images

The simplest and most obvious method of qualitatively code textual information is to read each document in a project and manually assign existing codes in a codebook to a selected segment of text. QDA Miner allows you to easily attach a code to a block of text that may be as small as a single character but that will more often consist of a few words, or one or more sentences or paragraphs. A code may even be assigned to an entire document.

- **Assigning Codes to Graphic Regions**
  - Select the area of the graphic you want to code.
  - Double-click the code in the code list.

- **Attaching a Comment to a Coded Segment**
  - Select the coded segment to which you want to attach a comment by clicking its code mark.
  - Click a second time to display the shortcut menu.
  - Select the COMMENT command. A small window like the one below will appear.

Analyze Menu

You will find that several to many of the same codes will be used repeatedly throughout. This is both natural and deliberate – natural because there are mostly repetitive patterns of action and consistencies in human affairs, and deliberate because one of the coder’s primary goals is to find these repetitive patterns of action and consistencies in human affairs as documented in the data. You search for patterns in coded data to categorize them, understand that sometimes you may group things together not just because they are exactly alike or very much alike, but because they might also have something in common – even if, paradoxically, that
commonality consists of differences. Briefly codes are not simple containers of information, they get their meanings within a structure where they are connected in terms of some relations. The aim of qualitative analysis is to reveal the structure and the relations that connect that structure.

QDA Miner provides several tools to assist in the coding task and to perform descriptive, comparative and exploratory analysis of codings. These tools may be used to systematize the coding of documents, ensure coding consistency, identify regularities and patterns in coding, uncover hidden relationships between codes and other properties of the cases.

Text Retrieval

The TEXT RETRIEVAL function searches for specific text or combination of text in documents. You can search in all documents in a project or restrict the search to specific document variables. Searches can also be restricted to specific coded segments.

*Analyze>*Text Retrieval

Write the words or phrases being searched for into *Search Expression* box. Boolean operators **AND**, **OR** and **NOT** may also be used to build complex search expressions.

- **Performing thesaurus-based searches**
  A thesaurus-based search allows one to search for several words or phrases associated with a single thesaurus entry previously defined. For example, by entering a single category [@GOOD (with an ampersand character "@") as a prefix), the program can automatically search for items that have been associated with this category, like "good", "fine", "excellent", "all right", "topnotch", etc. Names of categories may be typed directly in the search expression by preceding the category with a # character. For example, the following text search expression: @GOOD and @SERVICE will retrieve all text units containing either one of the words or phrases associated with the thesaurus entry GOOD along with any word or phrase included in the SERVICE thesaurus entry.

The Thesaurus Editor can be accessed from the Text Search dialog box by clicking the icon located next to the text search expression.
To create a new category:
Click the NEW button. A dialog box similar to this one will appear.

You can enter the expression you are searching for and find and copy and paste the synonyms from a ready-made thesaurus like Webster’s.

You can add new categories, edit and delete existing categories.

Query by Example
The QUERY BY EXAMPLE search tool is an alternative to a keyword-based search. Rather than specifying a long list of keywords joined by Boolean operators, this query method looks for documents or text segments similar to a given example and presents them in an initial list of search hits in descending order of similarity. The user then marks specific items as relevant or irrelevant and then performs a new query. The system adjusts the initial query by taking into account the provided relevance information and often provides better search results. After a single or a few iterations, the user can then select relevant items for further processing.

Analyze > Query by Example
The Starting Example section allows you to choose whether the starting example will be a Sample Text typed or pasted from the clipboard in an edit box or whether it will be based on existing Coded Segments associated with one or several codes. You’ll be asked for non-examples as well as examples. It may be a good idea to provide non-examples that are close in content to the starting examples. For example, if you are looking for paragraphs about globalization of economy, providing only globalization examples may retrieve paragraphs that are associated not only with globalization but also paragraphs containing words related to economy or foreign policy (names of other countries). If you provide non-examples, sample text segments about local economy or foreign policy that are not related to globalization per se, the program will apply less weight to words more specific to the economy and foreign policy and less related to the topic of globalization and more weight to words specifically about globalization.

FUZZY STRING MATCHING under the Options menu allows one to match documents even if they do not necessarily share the exact same relevant words. Computation of similarity is normally achieved by looking at shared words and requires exact matches of those words. Such an approach can perform poorly on documents containing misspelled words and may also fail to retrieve relevant documents using slightly different word forms. If the starting example has the word "aggression", QDA Miner may fail to match text segments containing words like "aggressive" or "aggressiveness" and will also fail to match other segments containing misspelled forms such as "agression". Enabling the Fuzzy String Matching algorithm relaxes the matching condition for words and retrieves more text segments than are likely to be relevant. Please note however, that this option may also increase the number of irrelevant results retrieved.

Click Search button:
This feature may be useful for quick top-down coding of documents, especially if you have a codebook and large amount of documents. You may not have to read all documents but may retrieve only the relevant parts and code.

To assign a code to a specific search hit:
- In the table of search hits, select the row corresponding to the text segment you want to code.
- Use the CODE drop-down list located above this table to select the code you want to assign.
- Click the button to assign the selected code to the highlighted text segment.

To assign a code to all search hits:
- Use the CODE drop-down list located above this table to select the code that you want to assign.
- Click the button to assign the selected code to all text segments meeting the search expression.
- Prior to the assignment of a code to all search hits, you may want to remove selected hits that do not correspond to what you were looking for. To remove a search hit from the list, select its row and then click the trash button.

Section Retrieval
The SECTION RETRIEVAL function searches structured documents for sections bounded by fixed delimiters. This feature is especially useful to automatically assign codes to recurring sections within a document or in multiple documents. For example, to retrieve the parts of interview scripts concerning only the “Therapist” you can ask to retrieve parts of the document starting from just after the title Therapist and finishing before the title “Client”

Assigning codes to the retrieved sections is same as the Query by example section. The coded segments can later be used for creating new cases (such as Therapist and Client) as explained in the “case splitting” section.

Keyword Retrieval
The KEYWORD RETRIEVAL feature can be used to retrieve any document, paragraph, sentence, or coded segment containing a specific keyword or a
combination of keywords associated with a QDA Miner code or stored in a WordStat content analysis dictionary.

We have already seen how to enter keywords during adding a code. Keywords may also be stored externally in categorization model files. Categorization models are advanced content analysis processes created by WordStat 5.0 or later and saved to disk in a .wcat file. A WordStat categorization model may involve various text processing such as stemming, lemmatization and word exclusions. It also involves the categorization of words, word patterns, and phrases and may also include complex coding rules involving boolean and proximity operators. Such rules may be used to perform disambiguation of words or coding of complex actions. In order to be used within QDA Miner, categorization models should be stored in a Models subfolder located under the main program folder.

**How to create a .wcat file:**

WordStat can categorise words, word patterns, phrases and coding rules into content categories using a categorization dictionary. You can either use custom built categorisation models (coding schemes, dictionaries) according to a theory such as the Harvard IV-4 dictionary, the Lasswell value dictionary (for more information on these dictionaries refer to general inquirer project: [http://www.wjh.harvard.edu/~inquirer/homcat.htm](http://www.wjh.harvard.edu/~inquirer/homcat.htm)). You can also use thesauri like Merriam-Webster or Roget. You can find some of these dictionaries at: [http://www.provalisresearch.com/wordstat/RID.html](http://www.provalisresearch.com/wordstat/RID.html). This webpage includes some theory based categorisation models such as Regressive Imagery Dictionary (RID) or thesauri such as Roget. Or you can also create your own categorisation models according to your research question, theory or inductively through pattern detection techniques such as cluster analysis (we’ll see more on this at the WordStat section).

You can use these categorisation models for a semi-automatic coding of the documents. Keyword retrieval feature is an excellent tool for bridging qualitative and quantitative content analysis coding procedures. For example, let’s assume that you are using RID as your coding frame. Normally you can perform an automatic content analysis through WordStat. However, automatic coding procedures take into account only the denotative aspects of the meaning units. Words get their meanings within a context; their connotation changes according to context. Understanding connotations is an interpretive effort and thus requires human intervention. Keyword retrieval feature can be used to see the keywords related to a category or code connote the expected meaning of that category. In RID, the category ANXIETY is denoted with words like worry, fear, trouble, crisis etc… However, these words may not always necessarily denote to ANXIETY. For example, a slogan like *no-fear* denotes to Nike brand rather than to ANXIETY within the context of advertisements. Keyword retrieval retrieves all the passages coded automatically as ANXIETY and allows a human reading and revision of the coding. So, when the human reader comes across the slogan *no-fear* among the retrieved words automatically coded as ANXIETY, s/he will recognise this and recode as NIKE rather than anxiety.
To perform these operations first we need to create a categorisation model and save it as a .wcat file:

- Create your categorisation model within the WordStat (refer to WordStat section) or open a custom built categorisation model like RID.
- Go to the Frequencies page and click on the button located at the top right hand side of the page (fifth among a series of buttons).
- Select the Export Categorization Model command. A dialog box should appear asking you for a file name.
- Enter the file name of the model you want to create and click Save. By default, categorization model files are saved with a .wcat file extension in the \Models subfolder under the program folder. Keep it that way.

To retrieve the passage coded with the keywords:
Analysis>Keywords
Select your Categorisation model. If keywords have been stored with the current project codebook, the list will also include the <Internal keyword> item. These are the keywords created during the codebook management process. You can choose an external categorisation model saved as .wcat file.

Set which document variables (i.e.; Speech) and search units (paragraphs in this example) the keywords would be searched in and a keyword filtering that includes the keywords you are looking for. Click on Search. To see the Search hits, click on it.
When searching for paragraphs, sentences or coded segments, the table also displays the text associated with the retrieved unit and its location (its paragraph and sentence number). Tables may be printed, stored as a text report, or exported to disk in various file formats such as Excel, ASCII, MS Word, HTML or XML.

In our example, search hits include the paragraphs involving the words denoting ANXIETY. A human coder can review these paragraphs to see the word in the context and decide if it connotes to the right keyword, i.e., ANXIETY.

*If needed, the paragraphs can be re-coded by applying the procedures for assigning new codes to the search hits.*

**Coding Frequencies**

The most basic analysis in text analysis is the simple count of words or themes. The assumption made is that the words or themes that are mentioned most often are the ones that reflect the greatest concerns. You can use the CODING FREQUENCIES command from the ANALYSIS menu to obtain a list of all codes in the current codebook along with their description and the category to which they belong. This dialog box may also be used to obtain various statistics for each code such as their frequency, the number of cases in which they are found, and the total number of words in the associated text segments. You can view the frequencies in terms of tables and charts to see the greatest concerns. The procedures for these are quite intuitive for an experienced social science software user; please refer to the Manual for further details.

For example, in the presidential candidate example most occurring theme is “Patriotism”; we can see the salience of the themes from the table below:

<table>
<thead>
<tr>
<th>Category</th>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>% Cases</th>
<th>% Words</th>
<th>MB Words</th>
<th>% Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>TocStwerin</td>
<td>33</td>
<td>15.2%</td>
<td>16</td>
<td>83.3%</td>
<td>3488</td>
<td>12.4%</td>
<td></td>
</tr>
<tr>
<td>TocStwerin</td>
<td>32</td>
<td>12.0%</td>
<td>10</td>
<td>61.1%</td>
<td>3601</td>
<td>11.9%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>7</td>
<td>3.1%</td>
<td>6</td>
<td>66.0%</td>
<td>3700</td>
<td>12.8%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>15</td>
<td>7.1%</td>
<td>6</td>
<td>56.0%</td>
<td>1809</td>
<td>7.1%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>18</td>
<td>7.1%</td>
<td>6</td>
<td>66.0%</td>
<td>1815</td>
<td>7.1%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>14</td>
<td>6.5%</td>
<td>6</td>
<td>66.0%</td>
<td>1110</td>
<td>4.1%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>11</td>
<td>4.5%</td>
<td>6</td>
<td>41.7%</td>
<td>767</td>
<td>2.8%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>19</td>
<td>9.5%</td>
<td>7</td>
<td>55.8%</td>
<td>899</td>
<td>3.2%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>9</td>
<td>4.5%</td>
<td>6</td>
<td>33.1%</td>
<td>999</td>
<td>3.5%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>7</td>
<td>3.4%</td>
<td>3</td>
<td>20.6%</td>
<td>511</td>
<td>1.8%</td>
<td></td>
</tr>
<tr>
<td>ValStwerin</td>
<td>5</td>
<td>2.5%</td>
<td>3</td>
<td>20.6%</td>
<td>165</td>
<td>0.4%</td>
<td></td>
</tr>
</tbody>
</table>

**Coding Retrieval**

The CODING RETRIEVAL feature lists all text segments associated with some codes or patterns of codes. Coded segments corresponding to the search criteria are returned in a table on the Search Hits page. You can either make simple searches or complex searches involving a combination of codes defined using one of seven logical operators (refer to the Manual for these operators). These logical operators may be very useful for detecting patterns of varying forms.

Hatch (2002) offers that you think of patterns not just as stable regularities but as varying forms. A pattern can be characterized by:

- similarity (things happen the same way)
- difference (they happen in predictably different ways)
- frequency (they happen often or seldom)
- sequence (they happen in a certain order)
- correspondence (they happen in relation to other activities or events)
- causation (one appears to cause another) (p. 155)

Complex searches with logical operators can help us to reveal these characteristics of code patterns related in various forms. For example, if you are looking for statements such as “we need to be powerful to protect our freedoms and power is justified as an ethical obligation and as a globalisation imperative”; you can search for text segments where the Power theme occurs near the text segments coded as Freedom (similarity) and overlaps with Ethic and Globalization (correspondence):

The retrieved codes are displayed in a table found on the Search Hits page.

Beside the text associated with retrieved coded segments, this table also contains the code name along with its category, the case from which this segment originates, the coder name and the date when it has been assigned. The Words column contains the total number of words in the text segment, while the %Words column shows the relative importance of this segment versus the entire document, in term of percentage of words. If a comment has been assigned to a specific segment, it will also be included in the table. Lastly, values for all variables that you have chosen to add to the table will be listed at the right end of the table.

To sort this list of retrieved segments in ascending order on any column values, simply click this column header. Clicking the same column header a second time sorts the rows again in descending order.
Coding Co-occurrences

QDA Miner allows you to further explore the relationship among codes by providing various graphic tools to assist the identification of related codes. These tools are obtained through the computation of similarity or co-occurrences index and the application of hierarchical cluster analysis and multidimensional scaling on all or selected codes. Results are displayed in the form of dendrograms, concept maps and proximity plots. Cases may also be clustered based on their content similarity using the same statistical and graphic tools.

This feature may be used to detect the patterns connected in terms of “similarity” or “overlapping” relations (remember Hatch’s list).

- **Options page:** is used to restrict the analysis to specific document variables or specific codes. It specifies whether clustering should be performed on codes or on cases and can be used to set various analysis and display options for both enters of analysis.

You can either choose to **Cluster Cases or codes**

- **Clustering cases**
  When the clustering is set to be performed on cases, the distance matrix used for clustering and multidimensional scaling consists of cosine coefficients computed on the relative frequency of the various codes. The more similar two cases will be in term of the distribution of codes, the higher will be this coefficient.
  
  **Data button:** sets the decision criteria on which this computation will be based.
  - **Code Occurrence:** based on which codes appear in each case, without taking into account the number of times each code appears.
  - **Code Frequencies:** takes into account the number of times a code has been used.
- **Number of Words**: comparison on the total number of words that have been assigned to different codes,
- **Percentage of Words**: option will divide this value by the total number of words found in documents associated with each case.

- **Clustering Codes**
  When clustering codes, several options are available to define co-occurrence and select which similarity index will be computed from the observed co-occurrences.

- **Occurrence** - This option allows you to specify how a co-occurrence will be defined. By default, a co-occurrence is said to happen every time two codes appear in the same document (within document option). You may also restrict the definition of co-occurrence to codes that are separated by no more than a specified number of paragraphs (window of n paragraphs option), or to codes that overlap each other (segment overlap option). Lastly, you can also restrict the definition of co-occurrence even further by limiting it to instances where codes have been assigned to the exact same segment.

- **Index** - The Index option allows the selection of the similarity measure used in clustering and in multidimensional scaling. Four measures are available. The first three measures are based on the mere occurrences of specific codes in a case and do not take into account their frequency. In all these indexes, joint absences are excluded.
  - **Jaccard's coefficient** - This coefficient is calculated from a fourfold table as \( a/(a+b+c) \), where \( a \) represents cases where both items occur, and \( b \) and \( c \) represent cases where one item is found but not the other. In this coefficient equal weight is given to matches and non matches.
  - **Sorensen's coefficient** - This coefficient is similar to Jaccard's but matches are weighted double. Its formula is \( 2a/(2a+b+c) \), where \( a \) represents cases where both items occur, and \( b \) and \( c \) represent cases where one item is present but the other one is absent.
  - **Ochiai's coefficient** - This index is the binary form of the cosine measure. Its formula is \( \sqrt{a^2/((a+b)(a+c)))} \), where \( a \) represents cases where both items occur, and \( b \) and \( c \) represent cases where one item is present but not the other one.
  - The last coefficient takes into account not only the presence of a code in a case, but also how often it appears in this case.
  - **Cosine theta** - This coefficient measures the cosine of the angle between two vectors of values. It ranges from -1 to +1.

- **Dendrogram**
QDA Miner uses an average-linkage hierarchical clustering method to create clusters from a similarity matrix. The result is presented in the form of a dendrogram:

Dendrogram can give us a nice summary of the patterns among the codes in terms of their co-occurrences. The idea is that the co-occurrence of codes implies a semantic similarity. Hence, we can accept clusters of codes as larger themes or topics which can be revealed from the internal organisation of the text. For example, codes such as Family, Patriotism, Tradition and Protectionism come together within a cluster suggesting a *conservatism* theme. Or Freedom and Power may imply a “we need to be powerful to protect our freedoms” theme. Evidently, naming of these themes is totally subjective and one needs to ground them back into the context by using various retrieval features previously mentioned. The patterns automatically revealed by the cluster analysis should be validated through rigorous interpretive pattern detecting which takes into account the context of the codes.

A word of caution though: the results of the cluster analysis are highly dependent on the similarity measure and the method of clustering. The results may change dramatically when one changes the measure and/or the method. Also, there are different criteria for deciding on the number of clusters. Reader may refer to a reference about cluster analysis for more detailed explanations.

QDA Miner’s cluster analysis features are limited but it allows to export the co-occurrence matrix to formats suitable for analysis in more sophisticated statistical packages such as SPSS.

- **2D and 3D Concept Maps**
  - The concept maps are graphic representations of the proximity values calculated on all included items (codes or cases) using multidimensional scaling (MDS). When clustering codes, each point represents a code and the distances between pairs of points indicate how likely these codes tend to appear together. In other words, codes that appear close together on the plot usually tend to occur together, while codes that are independent from one other or that do not appear together are located on the chart far from each other. The interpretation of multidimensional scaling plot is somewhat different when analyzing cases. In this case, each point represents a case and the distance between pairs of points indicate how similar the two cases are. Cases with similar patterns of codes will tend to appear close each other, while dissimilar cases will be plotted far from each other. Colours are used to represent membership of specific items to different partitions created using
hierarchical clustering. The resulting maps are useful to detect meaningful underlying dimensions that may explain observed similarities between items.

- **Proximity Plot**
  Cluster analysis and multidimensional scaling are both data reduction techniques and may not accurately represent the true proximity of codes or cases to each other. In a dendrogram, while codes that co-occur or cases that are similar tend to appear near each other, one cannot really look at the sequence of codes as a linear representation of these distances. You have to remember that a dendrogram only specifies the temporal order of the branching sequence. Consequently, any cluster can be rotated around each internal branch on the tree without in any way affecting the accuracy of the dendrogram. While multidimensional scaling is a more accurate representation of the distance between codes, the fact that it attempts to represent the various points in a two- or three dimensional space may result in distortion. As a consequence, some items that tend to appear together or to be very similar may still be plotted far from each other.

When looking at coding co-occurrences, selecting an item enables the button. Just selecting the code from the drop down Menu retrieves every pair of coded segments co-occurring, allowing one to further explore the factors that may explain this co-occurrence.

The proximity plot is the most accurate way to graphically represent the distance between objects by displaying the measured distance from a selected code to all other codes on a single axis. It is not a data reduction technique but a visualization tool to help you extract information from the huge amount of data stored in the distance matrix at the origin of the dendrogram and the multidimensional scaling plots.

In this plot, all measured distances are represented by the distance from the left edge of the plot. The closer a code is to the selected one, the closer it will be to the left.

Switching to the Table mode allows you to examine in more detail the numerical values behind the calculation of these plots. When the distance measure is based on co-occurrences, the table provides detailed information such as the number of times a given code co-occurs with the selected one (CO-OCCURS) and the number of times that it appears in the absence of this...
selected code (DO NOT). This table also includes the number of times the selected code appears in the absence of the given code (IS ABSENT).

- **Statistics:**
  Finally, the Statistics window reproduces the co-occurrence or similarity matrix which can be saved as an Excel file for further analysis for other purposes. For example, this matrix can be used for a network text analysis by using a software such as UCINET. One can also use this matrix for further statistical analysis in more sophisticated statistical software such as SPSS.

![Statistics Window](image)

**Code Sequence Analysis**

While coding co-occurrences analysis looks at the concomitant presence of codes regardless of their order of appearance in the documents, the Code Sequences finder is used to identify codes that not only co-occur, but do so in a specific order and under specific conditions. This command lists the frequency of all sequences involving two selected sets of codes as well as the percentage of time one code follows or is followed by another one. This can give us significant information about the argumentation structure within a text corpus. Since statements are the building blocks of a line of argumentation and two consecutive related concepts form a statement, the patterns in code sequences can give us important information about the structure of argumentation within a text corpus.

This feature may be used to detect the patterns connected in terms of “sequence”, “causation” or “difference” relations (remember Hatch’s list).

**FIRST CODES and SECOND CODES** - These two options allow you to select which specific code sequences will be analyzed. By default, the analysis is performed on all codes in the codebook, providing you with a comprehensive list of all sequences found in the documents. The list of sequences can however be limited by restricting either the leading or the following codes. For example, if someone is interested in the reactions of a therapist to client verbalizations, he may set the First Codes option to include only the codes used to describe client verbalizations and limit the codes in the Second Codes option to those used to describe the therapist's reaction. You can select specific codes for these two options either from the corresponding drop-down checklist by clicking the arrow button.
MINIMUM DISTANCE - This option is used to specify if a partial overlap will be allowed or not.

MAXIMUM DISTANCE - This option is used to specify what should be the maximum distance separating two codes in order to consider them to form a sequence.

After clicking the Search button, if at least one sequence if found, the Frequency Matrix and Search Hits page become enabled.

**Frequency Matrix**

The Matrix page shows a matrix of all code sequences found in the document. The starting code (code A) in the A-B sequence is shown on the rows, while the code that follows (code B) is displayed at the columns.

By default, the matrix displays the Z values of all sequences. In this example, colours are used to represent deviation from the expected frequency. Cells in green represent sequences occurring more often than expected, while cells in red represent sequences occurring less often.

The Show list box allows you to display other statistics. The following sequence statistics are available:
- Frequency of A followed by B
- Frequency of B followed by A
- Percentage of A followed by B
- Percentage of B followed by A
- Percentage of sequences equal to AB
- Z-value
- Probability
Search Hits

The **Search Hits** page allows you to examine in more detail the text segments associated with specific code sequences.

The occurrence of two codes within a sequence implies a relation but cannot tell us much about the nature of this relation. This relation may be an association ("sequence"), "causation" or opposition, contrast ("difference"). To detect this, we need to ground the relation in the context. **Search hits** feature provides a way to examine the retrieved segments in their surrounding context.

**Coding by Variables**

The **CODE FREQUENCY** dialog box is used to explore the relationship between codes assigned to documents and subgroups of cases defined by values of a numeric or categorical variable. When performing these kind of analyses, QDA Miner displays a contingency table containing either codes frequencies, code occurrences (present or absent), or either absolute or relative importance (in number or percentage of words). This dialog box also gives you access to several graphical and statistical tools to help you assess or visualize the strength of the relationship between code usage and the selected variable. This contingency table can be used to see the relations between the values of a categorical variable (like candidate, or political tendency, gender, age group etc…) and codes.
**CODES** - This option allows you to select which codes will be analyzed. By default, the analysis is performed on all codes in the codebook.

**TABULATE WITH** - This option list all numeric and categorical variables available in the project. Selecting any variable name will display a contingency table so that you can assess the relationship between this variable and the codes assigned to text segments.

**COUNT** - **Code Occurrence** allows you to compare the number of cases in which at least one instance of this code appears. If a code is used more than once in the same case, it will be counted only once. To count the total number of times a code has been used, select **Code Frequency**. Selecting **Word Count** will retrieve the total number of words that have been assigned to a specific code, while the **Percentage of Words** option divide this value by the total number of words found in those documents.

**DISPLAY** - The DISPLAY list box allows you to specify the information displayed in the table.

**STATISTIC** - This list box allows you to choose among 11 association measures to assess the relationship between the numeric or categorical variable and the codes.

*After a first crosstabulation table has been obtained, additional buttons located on top of this dialog box will become available. These buttons are used to create barcharts or line charts representations of data in selected rows, perform correspondence analyses, or create heatmaps. You can also export the crosstabulation table may be saved to disk in Excel, MS Word, plain ASCII, text delimited, HTML, or XML for further analysis with other software such as SPSS.*

- **Barchart or Line Chart**

- **Heatmap Plot**

Heatmap plots are graphic representations of crosstab tables where relative frequencies are represented by different color brightnesses or tones and on which a clustering algorithm is applied to reorder rows and/or columns. This
enter of plot is commonly used in biomedical research to identify gene expressions. When used in qualitative analysis, this exploratory data analysis tool facilitates the identification of functional relationships between related codes and group of values of an independent variable by allowing the perception of cells clumps of relatively high or low frequencies or of outlier values.

○ **Correspondence analysis**
Correspondence analysis is a descriptive and exploratory technique designed to analyze relationships among entries in large frequency crosstabulation tables. Its objective is to represent the relationship among all entries in the table using a low-dimensional Euclidean space such that the locations of the row and column points are consistent with their associations in the table. The correspondence analysis procedure implemented in QDA Miner allows you to graphically examine the relationship between assigned codes and subgroups of an independent variable. The results are presented using a two- or three-dimensional map. Correspondence analysis statistics are also provided to assess the quality of the solution.

Codes closely associated with the values of a categorical variable (candidates in this example) will be plotted at an angle from the origin that will lie between those values groups. In the above example, Power and Ethic discourses seem to be characteristic of Bush and McCain.
For a more complete description of this method, its calculation and applications, please refer to statistical reference. You can also export the crosstabulation to a statistical software such as SPSS for finer grained analysis.

- Other options include saving the cross-tab and printing.

Assessing Inter-coders Agreement

When document coding is performed manually by human coders, individual differences in interpretation of codes between different coders often occur no matter how explicit, unambiguous and precise the coding rules are. Even a single coder is often unable to apply the same coding rules systematically across time. One way to ensure the reliability of the application of coding rules is to ask different coder to code the same content or to ask a single coder to code the same document at different times. The comparison of codes is then used to uncover differences in interpretation, clarify ambiguous rules, identify ambiguity in the text, and ultimately quantify the final level of agreement obtained by these coders. Unfortunately, the application of inter-coders agreement procedures often involves some requirements or assumptions that are not always compatible with the qualitative data analysis processes. At least two compatibility problems can be identified:

1. The Codebook Problem - Computing inter-coders agreement requires that all coders share the same codebook. They should not be allowed to create their own codes or modify existing ones. An entirely inductive approach that would give each coder the possibility of developing his own codes during the analysis would result in a situation where the end result would very likely be incommensurable, each coder ending up with entirely different categories. In this situation, it would be impossible to establish the inter-raters agreement per se. Probably one of the only types of agreements that may be achieved in this case is by judging whether their final conclusions agree. While using a predefined unmodifiable codebook is the most obvious solution to this problem, a mixed solution may also be adopted. For example, the first coder may be given the opportunity to develop his own codebook or refine an existing one by adding categories or revising existing ones. Once the coding by the first coder is finished or once he is satisfied with the codebook state of development, this codebook can then be used by other coders to code existing documents. Another possibility would be to have a codebook that would contain a portion of unmodifiable codes but leave coders the possibility of modifying another portion of them or add new ones. In the latter case, only the fixed portion of the codebook could be submitted to an assessment of intercoders agreement.

I personally prefer the researcher to inductively develop the codebook and provide the coders with a predefined unmodifiable codebook.

2. The Segmentation Problem - Another problem is related to the common absence of predefined segments to be coded. The flexibility provided by many computer qualitative data analysis software to define the length and location of a segment to be coded creates a situation where the number of coded segments, their lengths and locations will very likely differ from one coder to the next. In this situation, a strict agreement definition that compares the assignment of the same codes with the same
text segment would be too stringent to apply. The most common solution to this segmentation problem has been the prior segmentation of documents to be coded. However, an alternative solution is to somewhat relax the definition of the agreement.

These problems can only be handled during the construction of the codebook phase. If the codebook is not carefully constructed and the coders are not accordingly trained, measuring the inter-coder reliability is a vain effort. Central concern of any coding procedure is the comparability of its results. Hence, in principle, every coder should make the same decisions on the quantification and classification of the documents in a text corpus. Therefore, the instructions in the codebook have to be studied thoroughly and followed to the core by the coders. The coders should be trained and pass a test before the coding procedure starts. Only after that reliability of the coding procedure can be measured by Inter-coders Agreement.

QDA Miner provides some features to measure Inter-coders Agreement.

But first, in order to assess inter-coder reliability, some prior settings and conditions are required:

- The project should be configured to support login by more than one user (for more information on how to define multiple users, see Security & Multi-users Settings).

- Each user with coding rights should then log into the project using his own user name and password in order to apply codes to documents.

- To ensure the independence of the coding process, a specific user should not be allowed to view code assignments made by other users. We have already seen this in the The QDA Miner Multi-users Settings section. This feature provides the option of hiding all codings made by other users.

- Each user has to code a common portion of the documents using the same codebook. The agreement can only calculated on documents that have been coded by more than one user. All other documents that have been coded by only one user will be ignored in the computation of the inter-coders agreement.

- Computing Inter-coders Agreement

**Analyze>Coder Agreement**
Analysis page:
CODES - This option allows you to select which codes will be analyzed.
CODERS - This option allows you to select the coders for which you want to assess the coding agreement. If more than two coders are selected, the agreement table will be calculated on all possible pairs of coders provided that they have coded the same documents. By default, this option is set to include all coders.

AGREEMENT CRITERION –

- **Code occurrence** - This option checks whether coders agree on the presence of specific codes in a document, regardless of the number of times this code appears and of code location. The agreement is calculated on a dichotomous value that indicates whether the code is present or absent. Some coders may agree that something is present in a document without necessarily agreeing on what specifically are the manifestations of this "something". For example, a researcher may ask two coders to check numerous job interviews and ask them to decide whether there are some indications of anxiety from the candidate. They may very well agree in the identification of such anxiety despite their inability to agree on its specific manifestations. If a research hypothesis or an interpretation is related to the mere presence or absence of anxiety in various candidates, this level of agreement may well be sufficient. Coders do not necessarily have to agree on exactly where it occurs.

- **Code frequency** - This option checks whether coders agree on the number of times specific codes appear in a document. This criterion does not take into account the specific location of the codes in the document. The agreement is calculated by comparing the observed frequency per document for each coder. For example, if someone wants to examine how often a specific speaker uses religious metaphors in his speeches compared with other speakers, he may ask different coders to assign code "Religious Metaphor" every time they encounter one in a speech. Even if they disagree somewhat on what represents a religious metaphor, they may nevertheless agree on how many of those metaphors they find in each speech. Since the research question is related to the comparison of frequency of metaphors among speakers, then an agreement on the overall frequency per speech should be sufficient.

- **Code Importance** - This option allows you to assess whether coders agree on the relative importance of a code in the document. This kind of agreement is achieved by comparing the frequency of words in the document that have been tagged as instances of this. For example, if a researcher makes the hypothesis that when adolescents are asked to talk about a love relationship, male adolescents devote more time than females talking about the other person's physical characteristics and less time about their own needs or feelings. With this kind of hypothesis, the researcher needs to establish an agreement on the relative importance of these topics, not necessarily on the frequency or location. Again, coders may somewhat disagree on what specific segments represent instance where the adolescent is talking about their needs or feelings, but they may nevertheless come to a close agreement on the importance of each code.
• **Code Overlap** - This last level of agreement is the most stringent criterion of agreement since it requires the coders to agree not only on the presence, frequency and spread of specific codes, but also on their location. When this option is selected, you need to provide a minimum overlap criterion expressed in percentage. By default, this criterion is set to 90%. If the codes assigned by both users overlap each other on 90% of their length, then these codes will be considered to be in agreement. If the percentage of overlap is lower than 90%, the program will consider them to disagree.

Such a level of agreement would be interesting for several reasons. A researcher may be interested in indentifying specific manifestations of a phenomenon. Another important reason is that the examination of agreements and disagreements at this level is accessory to the achievement of high agreement on any other lower level. In other words, close examination of where agreements occur or do not occur should allow one to diagnose the source of disagreement and take corrective actions to establish a shared understanding of what each code means for every coder. In a sense, checking the agreement at this level may be used in a formative way to train coders. Yet, the final summative evaluation of agreement may still be performed on a less stringent level, provided that this lower level matches the research hypothesis.

*The first three criteria* use documents as the unit of analysis and calculated for each document a single numerical value per code and coder. This numerical value represents either the occurrence, the frequency of the specific code or the percentage of words in the document assigned to it. This measure is then used to establish the level of agreement. For example, when assessing agreement on code frequencies, if a user assigns a specific code to a document four times while another one uses this code only once for the same document, then the program will add 0.25 to the total number of agreements and 0.75 to the total number of disagreements. The absence of a code in the document is considered to be an agreement. Since the unit of analysis is the document and the numerical value for each code lies between 0 and 1, each document has an equal weight in the assessment of agreement, no matter how many codes are found in these documents.

*However, when using code overlap* as the criterion for assessing agreement, the unit of analysis becomes the assigned codes. For this reason, the total number of agreements and disagreements can be higher than for the other three assessment methods. In this case, the absence of a code in the document is not considered to be an agreement.

**STATISTIC** - The simplest measure of agreement for nominal level variables is the proportion of *concordant* codings out of the total number of codings made. Unfortunately, this measure often yields spuriously high values because it does not take into account chance agreements that occur from guessing. Several adjustment techniques have been proposed in the literature to correct for the chance factor. Three of these are currently available in QDA Miner:

• **Free marginal adjustment** assumes that all categories on a given scale have equal probability of being observed. It assumes that coder decisions were not influenced by information about the distribution of the codes.
This coefficient is equivalent to the Bennett, Alpert and Goldstein's (1954) S coefficient, Jason and Vegelius's (1979) C Coefficient, and Brennan and Prediger's (1981) kn Index (Zwick, 1988).

- **Scott's pi adjustment** (Scott, 1955) yields results that are similar to those obtained using Cohen's Kappa (Cohen, 1971). Cohen’s Kappa is a commonly used statistic to measure the agreement between two individuals when two binary variables are attempts by them to measure the same thing. Kappa measures the percentage of data values in the main diagonal of the table and then adjusts these values for the amount of agreement that could be expected due to chance alone. Like the Kappa, pi does not assume that all categories have equal probability of being observed and considers that coders decisions are influenced by this information. However, unlike the Cohen Kappa, pi treats any difference among coders in this distribution as a source of disagreement.

- **Krippendorff's alpha** (Krippendorff, 2004) is similar to Scott's pi but applies a correction for small sample sizes. The nominal alpha exceeds pi by \((1-pi)/2n\) (where n is the number of units coded by the two coders) and asymptotically approximates pi as sample sizes become large.

NOTE: The expected frequencies that are displayed in the inter-rater agreement tables do not necessarily correspond to the expected frequencies used in the above correction techniques. Rather, they correspond to the values used in the computation of chi-square statistics used in contingency tables. However, those values coincide with those used in the computation of Cohen's Kappa and Krippendorff's r.

**List of Hits page:**

Page displays a list of all codings that were used for the assessment of inter-coders agreement.
The table lists the case descriptor, the code name as well as the coders' values used for the calculation of agreement. When assessing code presence, code frequency or code overlap, this value will be either Yes (for present) or No (for absent). When assessing code importance, these columns will contain the total number of words assigned to this code.

**Conclusion**

We are now done with the basic features of QDA Miner. You can switch to content analysis module WordStat through Analyze > Content analysis for automatic analysis of words and phrases found in documents. When used with QDA Miner, WordStat can perform these analyses on entire documents or on selected code segments. It may also perform simple descriptive analysis or let you explore potential relationships between words, phrases or categories of words and other numeric or categorical variables.

There are many benefits to combining qualitative and quantitative analysis techniques. For example, quantitative analysis may be useful as an exploration tool prior to qualitative coding by allowing one to identify subtle differences in word usage between subgroups of individuals, or to quickly find the most common themes, topics of phrases according to the patterns in word occurrences. Restricting the analysis to segments associated with specific codes may also be useful to identify potential words or phrases associated with those codes. You may then use the QDA Miner text retrieval tool to identify other segments to which this code may be assigned. Quantitative analysis may also be useful after qualitative coding has been performed. For example, you may try to validate the conclusions drawn from manual coding by comparing those conclusions with the results obtained from a quantitative content analysis. Qualitative codings may also serve as the starting material to develop and validate a categorization dictionary that will allow automatic categorization of documents or cases.

**Introduction to WordStat**

WordStat is an add-on module that can be used to automatically detect and analyze words and phrases. It is specifically designed to study textual information such as responses to open-ended questions, interviews, titles, journal articles, public speeches, electronic communications, etc. WordStat may be used for automatic categorization of text using a dictionary approach or various text mining methods (i.e.; cluster analysis) depending on word co-occurrences (like Alceste). WordStat can apply existing categorization dictionaries or may be used in the development and validation of new categorization dictionaries. When used in conjunction with manual coding, this module can provide assistance for a more systematic application of coding rules, help uncover differences in word usage between subgroups of individuals, assist in the revision of existing coding using KWIC (Keyword-In-Context) tables, and assess the reliability of coding by comparing automatic coding to manual coding. Like QDA Miner, WordStat has built-in text mining tools such as cluster analysis, correspondence analysis and Multidimensional scaling. But text mining possibilities are not limited to these functions; data produced by WordStat can be exported to other software such as UCINET or SPSS that are built for more specific purposes.
QDA Miner does not work by itself, it comes as integrated to either QDA Miner or SimStat (a statistical analysis program like SPSS).

**To open WordStat from QDA Miner:**

*Analysis>*Content Analysis*

**VARIABLES** - This option allows you to specify on which document variables the analysis will be performed. If the current project contains more than one document variable, you can choose more than one (default: all document variables are selected). To restrict the analysis to a few of them, click the down arrow key at the right of the list box. Select the one which is the concern of your analysis. By default, text analysis is performed on the entire document (*All Text* option). However, it is also possible to restrict the analysis to specific coded segments or exclude segments from the analysis by enabling the **Coded Segments** option.

**IN RELATION WITH** - This set of options allows you to specify whether the quantitative content analysis should consider potential differences or relationship with one or more quantitative or categorical variables. For example, you may select gender variable to see if the pattern of words used by men and women differ. Or candidate variable to see if word or theme occurrences in presidential speeches for different candidates follow specific patterns.

Once the options have been set, click the **OK** button to call WordStat.

**Preparing and Importing Data**

- **Preliminary Text Preparation**
  - Check spelling of documents
  - Remove hyphenation
  - Add or remove square brackets ([ ]) and braces ( { } ). Square brackets and braces have special meanings for WordStat. For example, braces are used to remove a section of the text that you don't want to process while square brackets may be used to restrict the analysis to specific portions of text. If these symbols are used in a text for other purposes, they should be replaced with other symbols.
• **Entering data:** You cannot directly enter data to WordStat. Data input should be done either through QDA Miner or must be entered to a spreadsheet like Excel and imported.

• **To enter data through QDA Miner just follow the procedures stated previously**

• **To enter data through Excel:**

  • **Formatting spreadsheet data**
  Evidently, data import is not limited to spreadsheets; a variety of input files from SPSS to Atlas-ti (check Manual) are supported but for the sake of simplicity only spreadsheet data will be explained here.

Most spreadsheet programs allow for entry of both numeric and alphanumeric data into cells of a data grid. QDA Miner can import spreadsheet files.

I would recommend to use spreadsheet option for data involving short texts as well as large amount of variables such as surveys with open ended questionnaires. For longer texts, you can directly enter data to QDA Miner as explained in the previous sections.

The selected range must be formatted such that the columns of the spreadsheet represent variables (or fields) while the rows represent cases. Also, the first row should preferably contain the variable names while the remaining rows hold the data, one case per row. SIMSTAT and QDA Miner will automatically determine the most appropriate format based on the data it finds in the worksheet columns. Cells in the first row of the selected range are treated as field names. If no variable name is encountered, SIMSTAT and QDA Miner will automatically provide one for each column in the defined range. When reading the data for analysis, all blank cells and all cells that do not correspond to the variable type (e.g., alphanumeric entries under a numeric variable, or a numeric value under a string variable) are treated as missing values.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gender</td>
<td>Age</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Importing Spreadsheet Files**

To import data from a spreadsheet:

- From QDA Miner, choose the NEW command from the PROJECT menu and then select IMPORT FROM AN EXISTING DATA FILE.
- Select the file format using the List File of Type drop down list.
- Select the file you want to import and click on the OK button.

For importing files in other formats please check the Manual.
Content Analysis & Categorization Process: Dictionaries Page

The most basic form of content analysis that WordStat can perform is a simple frequency analysis of all words contained in one or several text fields of a data file. However, WordStat offers several features that permit the user to accomplish more advanced forms of content analysis that may involve automatic categorization, different weighting of words, inclusion or exclusion of words based on frequency criteria, etc.

Dictionaries Page

Without further information, WordStat can perform a frequency analysis on each of the words encountered in the chosen alphanumeric fields. However, it is also possible to apply various transformations on the words before performing the frequency analysis. The first two pages of the main dialog box (see below) allow one to specify how the textual information should be processed. For example, you can tell the program to preprocess words using two dictionaries: an exclusion and an inclusion/categorization dictionary. The Dictionaries page also allows you to use or create those dictionaries, add, remove or edit existing entries in those dictionaries. For more information on how to open, activate or deactivate a dictionary or how to add, edit or remove an entry in a dictionary, see Creating and Maintaining Dictionaries, page 56).

- **To open an existing dictionary:**
  
  Go to the Folder icon at the right hand side of the Exclusion or Categorization buttons: you can browse for Exclusion (.txt files) or Categorization dictionaries:

- **To create a new dictionary:**
  
  Go to the icon at the right hand side of the Exclusion or Categorization buttons: you’ll be asked to name and save a new dictionary. If a dictionary is already open,
you’ll be asked if to keep or not the existing keywords. Do not, if you are creating a dictionary from the scratch. Follow the instructions.

Following steps are involved during the creation of a new dictionary:

- **PREPROCESSING**: The preprocessing option allows for the custom transformation of the text to be analyzed prior to, or in place of the execution of the other three standard processes provided by WordStat: lemmatization, exclusion and categorization of words and phrases. This transformation is accomplished by the execution of specially designed external routines accessible in the form of an external EXE file or a function in a DLL library. This feature customize the processing of textual information. For more information on this feature see Configuring External Preprocessing Routines in the Manual (p.80).

- **LEMMATIZATION**: The stemming process is used to reduce the various word forms to a more limited set of words or word roots. Such a process is typically used for lemmatization, a procedure by which all plurals are transformed into singular forms and past-tense verbs are replaced with present-tense versions. It may also be used for derivational stemming in which different nouns, verbs, adjectives and adverbs derived from the same root word are transformed into this single word. For more information on such a feature, see Monitoring and Customizing Lemmatization (p.77).

- **EXCLUSION OF WORDS**: An exclusion process may be applied to remove words that you do not want to be included in the content analysis. An exclusion dictionary (also known as a stop list) must be created to remove words with little semantic value such as pronouns, conjunctions, etc., but may also be used to remove some words used too frequently or with little discriminative value. WordStat has exclusion lists for major languages like English, French, Spanish and Italian. These lists are saved as .txt files and can be modified for special purposes. Users can also create custom exclusion lists for any language and save them as a .txt file.

- **CATEGORIZATION OF WORDS AND PHRASES**: The inclusion / categorization process allows one to change specific words, word patterns or phrases to other words, keywords or categories and/or to extract a list of specific words or codes. This process requires the specification of an inclusion dictionary. This dictionary may be used to remove variant forms of a word in order to treat all of them as a single word. It may also be used as a thesaurus to perform automatic coding of words into categories or concepts. For example, words such as "good", "excellent" or "satisfied" may all be coded as instances of a single category named "positive evaluation", while words like "bad", "unsatisfied" or expressions like "not satisfied" may be categorized as "negative evaluation". Inclusion dictionaries may also be used to perform a category based frequency analysis besides word based frequencies. A categorization dictionary may also contain rules delineating the conditions under which specific words or phrases should be categorized. Those rules may consist of complex expressions involving Boolean (AND, OR, NOT) and proximity operators (NEAR, BEFORE, AFTER). Those kinds of rules allow
one to eliminate basic ambiguity in words by taking into account the presence of other words that may alter the meaning. For example, the differentiation of the various meanings of the word BANK by identifying other words like "river", "money" and "deposit" surrounding "bank". For more information on rules, see section Working with Rules, p. 64.

The inclusion dictionary is structured as a hierarchical tree where words, word patterns, phrases, and rules are grouped in a folder that represents a category name. Categories and individual words may also be included in a higher order category, allowing one to create multi-level dictionaries. You can set the level of analysis through the Categorization button.

This option allows one to specify up to which level the coding should be performed. For example, in the following dictionary:

if a level of 1 is specified, all words that are stored at a higher level than the root level will be coded as the parent category at this first level. For example, words like CANADA and MEXICO will be coded as COUNTRY along with other country names like BRAZIL. Setting the level of analysis to a numeric value of 2 will results in the coding of those two words as NORTH-AMERICA, while BRAZIL will be coded as SOUTH-AMERICA. Items stored at the same or at a lower level than this option will remain unchanged. For more information on how to open, activate or deactivate a dictionary or how to add, edit or remove an entry in a dictionary, see Creating and Maintaining Dictionaries, page 56).

Options Page

This page offers different options that control how the textual information should be processed. The options are grouped under three different pages:
1) Analysis
2) Speller/Thesaurus
3) Others

Options menu are quite user friendly and intuitive. For further details refer to the Manual p. 28
The Frequencies page is used to display a frequency table of words or category names. This can be used to perform an univariate frequency analysis on words or categories and also to modify the dictionaries.

- The button can be used to move one or several words to the exclusion or substitution list or to add or remove a word from the inclusion list. It is also possible to quickly access the pop-up menu invoked by this button by pressing the right button of the mouse anywhere on the grid (see below).

- The button allows one to produce barcharts or pie charts to visually display the distribution of specific keywords or categories. Select the rows you would like to plot.
(multiple but separate rows can be selected by clicking while holding down the CTRL key), and click the button. For further information see Displaying Distribution Using Barcharts or Pie Charts on page 86.

- The button allows one to perform cluster analysis and multidimensional scaling on all included words or categories and display a dendrogram or concept map of those items based on their proximity. We have already seen this during the section on QDA Miner. For further information see Hierarchical Clustering and Multidimensional Scaling page 97).

- The options for dendogram feature are same as QDA Miner except the CLUSTERING TYPE option. Two broad types of keyword clustering are available.
  - The first method is based on keyword co-occurrences (First Order Clustering) and will group together words appearing near each other or in the same document (depending on the selected co-occurrence window).
  - The second clustering method is based on co-occurrence profiles (Second Order Clustering) and will consider that two keywords are close to each other, not necessarily because they co-occur but because they both occur in similar environments. One of the benefits of this clustering method is its ability to group words that are synonyms or alternate forms of the same word. For example, while TUMOR and TUMOUR will seldom or never occur together in the same document, second order clustering may find them to be pretty close because they both co-occur with words like BRAIN or CANCER. Second order clustering will also group words that are related semantically such as MILK, JUICE, and WINE because of their propensity to be associated with similar verbs like DRINK or POUR or nouns like GLASS.

- Using cluster analysis for bottom-up dictionary construction
  However, the logic of pattern detecting may be a little bit different then QDA Miner when we work with WordStat. Wordstat, does not depend on human coding of the documents. It is based on the occurrences of words or categories assigned to these words. There is less human involvement during the coding process. WordStat automatically counts the words or assign them to categories. Some software like Alceste claim to extract native categories inductively from the organisation of the text itself with minimum involvement of the researcher. The program generates an empirically based classification of text units according to the pattern of co-occurrences of word tokens within these units. These software depend on cluster analysis which is widely used by the cognitive anthropologists for detecting specific semantic systems and native categories embodying the patterns of everyday activities of a community. When studying text produced by different individuals, the aim is to understand points of view that are collectively shared by a social group at a given time. So these techniques may be used to inductively derive categorization dictionaries according to the word co-occurrence patterns in the text. Rather than
adopting a top-down, theory based approach to dictionary construction, a bottom-up, emic approach can be adopted.

To do this with WordStat, the analysis should start without a categorization dictionary since cluster analysis for detecting native categories will depend on word co-occurrences. However, steps for pre-processing the text like lemmatization, formation of exclusion dictionaries should preclude the cluster analysis. Another crucial point is restrict the definition of co-occurrences by limiting the co-occurrence to a window of sentences. Such an small window is especially useful when doing an analysis directly on words (rather than categories) since it allows to identify idioms or phrases that may need to be added to the categorization dictionary. Co-occurrence on larger text segments such as cases or paragraphs may be more appropriate to identify the co-occurrence of themes in individual subjects. For an example of bottom-up dictionary building through the cluster analysis of word co-occurrences and application of the categories recovered by the clusters as a coding frame for qualitative coding consider (Veltri and Suerdem 2011).

**Crosstab Page**

The Crosstab page is used to display a contingency table of words or categories. This contingency table is computed only on items that have been included. If an inclusion dictionary has been specified, this grid will display only the words or keywords in this list. If no inclusion list has been specified, the grid will display all words that have not been explicitly excluded. Along with absolute and relative frequency of keyword occurrence or keyword frequency, several statistics may be displayed to assess the relationship between independent variables and word usage or to assess the reliability of coding made by several human coders or a single coder at different times. This feature is similar to the QDA-Miner Coding by Variables feature.

**Keyword-In-Context Page**

The Keyword-In-Context (KWIC) technique allows one to display in a table the occurrences of either a specific word, or of all words related to a category, with the textual environment in which they occur. The text is aligned so that all keywords appear aligned in the middle of the table. This technique is useful to assess the consistency (or lack of consistency) of meanings associated with a word, word pattern or category. In the example below, we can see that the word pattern KILL*, which may have been assigned to a category like "aggressiveness", refers to words with different meanings, some of them quite distant from the concept of "aggressiveness":

*I have decided to KILL a few hours before...*

*He said that he would KILL me if I call the police.*

*Too much garlic KILL the taste of the meat.*

*The Black Death was a disease that KILLED millions.*
My shoes are KILLING me

The French skier Jean Claude KILLY won 3 gold medals.

When displaying rules, only the keywords or key phrases associated with the first item of those rules are displayed. For example, in a rule like:

#SATISFACTION before #TEACHER and not near #NEGATION

the KWIC list will contain only items in the SATISFACTION category meeting the conditions specified by this rule.

Once an inconsistency has been detected, it becomes possible to reduce it by making changes to the textual data or to the dictionaries. For example, the researcher may change all occurrences of the word KILL in the original text for either KILL1 or KILL2 in order to differentiate the different meanings and then add only one of these modified words (say KILL1) to the substitution or inclusion dictionary. The word KILLY may also be added to the dictionary of excluded words. The categorisation of phases may also be used to distinguish various meanings of a word. For example, the use of KIND to refer to the adjective ("considerate and helpful nature") may be reliably differentiate from the use of KIND as a noun ("category of things") or as an adverb by categorizing the phrase "KIND OF" as instances of this word used as a noun or as an adverb and by categorizing the remaining instances of KIND as the adjective. Disambiguation may also be performed by identifying words in close proximity that are associated with specific meanings and by creating categorization rules (see Working with Rules on page 64).

The KWIC technique is also useful to highlight syntactical or semantic differences in word usage between individuals or subgroup of individuals. For example, candidates from two different political parties may use the word "rights" in their discourses at the same relative frequency, but we may find that these two groups use this word with quite different meanings. We may also find that the meaning of a word like "moral" evolves with the age of a child.

- Clicking on the button produces a KWIC table:
• **Clicking on the button produces a concordance report on the keywords currently displayed in the KWIC table.** The sort order and context delimiter of the current KWIC table are used to determine the display order and the amount of context displayed in this concordance report. This report is displayed in a text editor dialog (see below) and may be modified, stored on disk in RTF, HTML or plain text format, printed, or cut and pasted to another application. Graphics may also be pasted anywhere in this report. Clicking this button produces a concordance report on the keywords currently displayed in the KWIC table. The sort order and context delimiter of the current KWIC table are used to determine the display order and the amount of context displayed in this concordance report. This report is displayed in a text editor dialog (see below) and may be modified, stored on disk in RTF, HTML or plain text format, printed, or cut and pasted to another application. Graphics may also be pasted anywhere in this report.

---

**Features Extraction Page**

The Features Extraction page provides two useful tools to extract relevant information from a text corpus:

• The Phrase Finder will extract the most common idioms, phrases and expressions in the text corpus.

• The Vocabulary Finder will identify proper nouns, company names and technical terms as well as common misspellings of words.

---

**PHRASE FINDER PAGE**

To accurately represent the meaning of a document, it is sometimes not good enough to rely on words alone but one should look at idioms and phrases. For example, the words “social” and “security” have different meanings by
themselves and a different meaning when they come together within the phrase: “social security”. While obtaining comprehensive list of words is easy, finding common phrases in a specific text corpus is often much more difficult. The phrase finder feature of WordStat provides such a tool. It will scan an entire text corpus and identify the most frequent phrases and idioms and allow one to easily add them to the currently active categorization dictionary. In order to reduce redundancy in such a table, short phrases that are part of larger ones are automatically removed from the list, provided that their frequency is lower than or equal to the frequency of a longer version.

To add an idiom or phrase to a categorization or exclusion dictionary click on the button.

Please note that because of the processing sequence used by WordStat, a phrase added to a categorization dictionary may still not be recognized if it starts with a word currently in the exclusion list. When WordStat encounters a word to be excluded, it automatically ignores further processing and moves to the next word. For this reason, a phrase should never begin with a word in the exclusion list. However, WordStat will recognize any other phrases containing but not starting with a word in this list. To override this and identify all possible phrases, go back to the Dictionary page and disable temporarily this exclusion list.

Finding overlaps

While WordStat tries to reduce redundancy in the list of phrases by automatically removing short phrases that are part of larger ones, the resulting list may still contain items that are not independent of each other such as phrases that sometimes overlap (such as “social security” and “social security reform”). In order to allow users to take into account potential overlaps when selecting phrases, WordStat provides a display option that allows one to see when a selected phrase includes a shorter one, is part of a larger one, or sometimes overlaps other phrases.
To enable the display of information regarding overlaps, simply click on the button. A window appears on the right of the table. Selecting a phrase in the table automatically shows all other items that overlap this selected item. Each phrase is accompanied by a ratio indicating the total number of times this other phrase occurs and how many times it overlaps with the selected item. For example, if one selects the phrase I'M LOOKING FOR in the table showing it occurs 26 times in a document collection, one may notice that it overlaps with another phrase, LOOKING FOR SOMEONE, with a ratio of 11 out of 12. This suggests that LOOKING FOR SOMEONE occurs 12 times, but on 11 occasions, both phrases overlap (I'M LOOKING FOR SOMEONE). This ratio also indicates that on one other occasion, this second phrase occurs without overlapping the first one. It is also useful to compare the total number of overlaps with the total frequency of the target phrase. In the above example, we can conclude that the phrase I'M LOOKING FOR – occurring 26 times - is followed by SOMEONE on 11 occasions. Thus, on 15 other occasions, it is followed by something else.

To hide information about overlaps, click on the button again.

• **Comparing frequencies or case occurrences of phrases**
  The distribution of a phrase among classes of a categorical variable may be quite useful when choosing whether or not to include it in a categorization dictionary. For example, one may want to identify phrases that are typical of some topics in order to better describe them or to differentiate them from other topics.

While the Crosstab page in WordStat is normally used for such a purpose, it can only be used for items already included in a categorization dictionary or selected by the content analysis process. In other words, one way to compare the frequency of phrases identified by the phrase finder among classes of a categorical variable is to move all those phrases to a categorization dictionary, and then use this dictionary to obtain the cross frequency of those phrases. However, the phrase finder page offers a convenient way to obtain such information without the need to move those phrases to the categorization dictionary.

To compare frequencies or case occurrences of phrases:

Once phrases have been extracted, click on the button.
• **Filtering the table**

Extracting phrases from a large collection of documents can result in a very large table containing thousands of phrases. Clicking on the button brings a dialog box offering filtering options that allow one to view only phrases containing either a key word or phrases that are characteristic of a specific class. Filtering conditions are specified in a dialog box similar to this one:

![Filter phrases dialog box](image)

• **VOCABULARY FINDER PAGE**

The Vocabulary Finder feature of WordStat provides a tool to extract single words representing technical terms, company and product names, as well as abbreviations that are specific to the analyzed collection of documents. The feature will also identify common misspellings by comparing the list of word forms encountered in the entire text collection against a list of common words, retrieving words or words with irregular capitalization that are not found in this list. By default, the extraction is performed in reference to common English words. For documents written in another language or to exclude technical terms from a specific domain, set the Active Dictionaries option on the Speller/Thesaurus Option Page to the desired language.

![Vocabulary Finder feature](image)

Three types of operations are allowed on these words: 1) You can replace all instances of a selected word in the original document by another word or phrase; 2) You can add this word to a custom list of valid words causing the program to ignore those words the next time there is a search for vocabulary words; or 3) You can assign the words to an existing
content category. You may also obtain a keyword-in-context list associated with a specific word in order to decide how that word should be treated. None of the above three operations are performed immediately. Instead they are added to an actions list allowing you to review, modify or cancel previously defined actions prior to the application of all the specified changes.
References and key readings


Montreal, Provalis Research.


You can also find an exhaustive list of studies used QDA Miner and WordStat including a variety of disciplines: