An Introduction to Qualitative Data Analysis Software Packages

Daniel Loewus-Deitch

Fielding Graduate University

An Introduction to Qualitative Data Analysis Software Packages

Computer-assisted qualitative data analysis software (CAQDAS) is a genre of computer applications that are designed for qualitatively analyzing text, images, audio, and video. CAQDAS packages allow researchers to code textual data by attaching classifiers and interpretive memos to segments of text. They also offer various data retrieval and analysis features, including multi-faceted searches, thesaurus building, and word frequency counts (Lewins & Silver, 2007).

Today, there are multiple CAQDAS packages to choose from. All of these application suites have grown into considerably mature products, adding both new features and improved user interfaces along the way. The major products in this category are NVivo, ATLAS.ti, MAXQDA, Dedoose, and Leximancer. The first three are longstanding desktop applications that have each had anywhere from 7-11 major releases. Dedoose and Leximancer are much newer to the scene; yet, they are both modern, feature rich, cloud-based web applications with some unique capabilities. These capabilities include sophisticated auto-coding, remote collaboration features, and support for mixed methods research approaches.

All CAQDAS software allows users to assign keywords, on the fly, to selected segments of raw data. Textual content can also be annotated in various ways, including the attachment of memos and comments. After coding, the database can be searched, filtered, sorted, and visualized in the form of conceptual mind maps. In addition, codes can be merged, split, and organized into structures, including hierarchies and families. Finally, researchers can use the software to discover relationships between codes by analyzing co-occurrences, proximity, word frequency, and other metrics (Lewins & Silver, 2007).

In this article, I will provide an overview of qualitative data analysis concepts, while discussing the various tasks that can be performed with CAQDAS software. I will then compare three CAQDAS solutions: Leximancer, MAXQDA, and Dedoose. Each of these software packages has significant strengths, as well as a few frustrating usability quirks.

Types of Qualitative Data

There are numerous types of qualitative data that can be analyzed in CAQDAS. Some of the more high-level categories are primary data, secondary data, supporting materials, and background information. Primary data often consists of interview transcripts, field notes, or participant responses to essay-style questionnaires. Secondary data includes newspaper articles, web forums, and official documents (e.g., government or business). Supporting materials can come from published academic literature, websites, television media reports, or any number of other sources. They provide domain knowledge, experimental results, and articulation of theoretical models. Finally, there is standard background information. This includes administrative correspondence between researchers throughout a research project, such as emails and meeting notes (Lewins & Silver, 2007).

Preparing Data for CAQDAS

Before importing data into CAQDAS, it can be helpful to prepare the text by formatting it in particular ways. It is helpful to come up with a consistent format and naming convention for both speaker and topic identifiers. This helps when using search tools within the software to find quotes from a particular speaker or instances of a certain topic. In addition, a common and meaningful naming convention for the data files, themselves, can be very useful for sorting and targeted retrieval of particular texts. Using rich text, such as colors, bolding, and italics can help one parse text while scanning and skimming through the documents. Lastly, collecting data in a

consistent, structured format when conducting interviews, focus groups, or observations can make subsequent analysis, inside the software, significantly easier (Lewins & Silver, 2007).

Qualitative Coding

At a basic level, qualitative coding involves selecting segments of raw data and tagging these segments with one or more categories. These segments may be words, phrases, sentences, or even multiple paragraphs of text. However, they may also be sub-areas of images or audio/video excerpts. Coding is a process of managing and ordering raw qualitative data (e.g., an interview or narrative) in a way that helps standardize and abstract it at a higher conceptual level (Lewins & Silver, 2007).

It is important to remember that coding is not necessarily a thorough analysis in and of itself, but rather it sets up the data to be analyzed in a more rigorous manner. It the first step in the analytical process. The goal of coding is to create a layer of meta-data, which makes the text easier to parse and search when later mining for relationships and reoccurring patterns that exist across the entire corpus of data collected. Codes also help researchers discover similarities and differences in the way ideas and concepts are expressed within the qualitative data (Lewins & Silver, 2007).

Manual Coding

Human researchers directly conduct manual coding. Coders read through the raw text, looking for instances of a conceptual pattern. When coders recognize a segment as an instance of a concept, they highlight the segment and link it to a code representing that concept. Segments can be sub-ranges of other, previously selected sections. Additional segments can also be created which overlap multiple ranges of data that have already been coded. Each highlighted segment may be tagged with as many categories as deemed relevant by the coders. Finally, the start and

end points of any segment can be modified at any time, based on later considerations (Lewins & Silver, 2007).

Codes do not always represent theoretical or abstract ideas. Often, codes are practical and descriptive in nature. They also can be markers of various types of language or terminology that is repeatedly used (Lewins & Silver, 2007).

Codes, themselves, may be created on the fly, as themes emerge from the data. This is the *inductive* approach. It is done from the bottom-up, based primarily on the Grounded Theory methodology. In Grounded Theory, one starts with no pre-determined hypotheses, and instead lets the theory or thematic patterns emerge organically from the data itself. Alternatively, codes may be developed a priori, based on an established theoretical framework or pre-determined hypothesis. In this *deductive* approach, the goal is often to test out an existing theory, using newly collected data. This also tends to be the preferred technique in applied settings, outside of academics. Here, the objectives of the research are usually much more targeted and narrow (Lewins & Silver, 2007).

Codes must always be carefully defined. Each code should contain specific details about its meaning and scope, as well as the way it is supposed be applied to data segments. Coders must clearly and consistently understand when and when not to use the category to tag a particular section (Lewins & Silver, 2007).

Regardless of the approach, coding is also not a finite task that is completed in a single pass-through of the data. Rather, it is an iterative process, in which there are multiple phases. Throughout the progressive phases, codes can be merged, split, renamed, redefined and even deprecated. Brand new codes can also be generated at any time. As the qualitative analysis continues, the data may be classified and recombined in a multitude of ways, until the

researchers are finally satisfied with their overall understanding and interpretation (Lewins & Silver, 2007).

Coding schemes and structures.

Once a list of codes is created, it is usually helpful to place them into some sort of structure. This is a crucial step in producing a useful coding scheme. Codes can be organized into either a hierarchical system or a non-hierarchical system.

Placing codes into a functional hierarchical system can often make researchers feel more organized. It is a particularly good option when using a deductive approach. However, it also can be quite beneficial when using Grounded Theory techniques to engage in inductive coding. A hierarchy is a great way to begin combining initial codes into higher-level themes during the axial and selective coding phases. Hierarchies have the potential to become problematic, however, if they grow beyond three levels. In this case, individual codes may become lost and forgotten about, and the hierarchy then becomes more of a disruptive distraction than a useful structure (Lewins & Silver, 2007).

In a non-hierarchical configuration, codes remain listed all at the same level. Prefixes can be used in the code names, themselves, in order to group the concepts via sorting. Some CAQDAS offerings also allow codes to be explicitly linked together as members of the same family or set. This is similar to creating an affinity diagram. A non-hierarchical system provides more visibility to all codes, because they are not hidden underneath collapsed categories. It also helps researchers focus more on relationships between codes that are not necessarily hierarchical in nature. Of course, the disadvantage of a non-hierarchical structure is that seeing all codes in one single level may be overwhelming, and it makes it harder to mentally abstract all of these codes up to a higher level (Lewins & Silver, 2007).

In the end, the structure of a coding scheme should effectively support your methodology. Codes may be renamed and reorganized within the structure numerous times throughout a project. If a coding scheme becomes too large and unmanageable, redundant codes can be merged. Rarely used codes can be deactivated or put aside into a separate group. Visual mind maps can also be generated in order to explore the coding frame and view the relationships in a different way. Finally, memos may be written and attached to codes. In these memos, researchers can document their detailed thoughts and concerns about the way codes are grouped or linked together (Lewins & Silver, 2007).

Auto Coding

In qualitative research, intimate immersion into the dataset is important. However, this can be an extremely time-consuming task, and it is prohibitively intensive for large bodies of qualitative data. This is where auto-coding comes in (Penn-Edwards, 2006).

Leximancer is the most advanced CAQDAS software package when it comes to auto-coding. It uses sophisticated algorithms to search and mine massive volumes of text (Penn-Edwards, 2006). Leximancer does a lot more than simple keyword searches. It can build a dictionary directly from the textual data itself and then extract thesaurus-based codes or concepts. When auto-coding, Leximancer goes through two extraction stages: semantic and relational (Smith & Humphreys, 2006).

During the semantic stage, Leximancer creates a dictionary of categories, learning the emergent, characteristic language that the textual body contains. Then, the classifiers are used to code each text segment, based on both frequency and co-occurrence of the terms throughout the document(s). Leximancer's engine examines how often two words appear in close proximity to each other, and also how often they appear far apart within the text. In addition, the researcher

can optionally seed meaningful terms into the dictionary before auto-analysis begins. In this case, the software will discover a set of concepts within the text that is related to the pre-inserted concepts (Smith & Humphreys, 2006).

During the relational stage, frequency and co-occurrence of categories are further analyzed in order to determine various relationships between the concepts. Based on these calculations, groups of concepts, or 'themes,' are formed. This helps produce a certain hierarchy within the code scheme, and ultimately a visual concept map is automatically generated (Smith & Humphreys, 2006). This map can be dynamically manipulated to show broader themes and/or more detailed concepts.

Auto coding is best used as a first step or phase in the analytical process. It uncovers a set of higher-level concepts that are present in the data, based on mathematical patterns of frequency and co-occurrence. It is important to remember, however, that these auto-coded categories are purely based on linguistic properties. This means that they don't represent style or the tone of the speaker (Penn-Edwards, 2006). With that said, Leximancer does have a "sentiment lens" feature which seeds a predefined list of common terms that relate to sentiment (Leximancer Manual, 2011). Regardless, in order to truly understand, interpret, and connect with the data, one must move beyond the initial concept map that a software application like Leximancer generates, and delve directly into the segments that have been auto-coded. Active immersion and exploration of the raw text is just as critical, whether the coding begins with a manual or an automated process.

Data Management

Data may be managed and organized in any number of ways, including its origin (e.g., a particular interviewee), its type (e.g., conversational dialog, newspaper article), or based on socio-demographic categories. It helps to determine, from the outset, any factual dimensions that

you desire to compare, focus on, or inquire about. Data may be grouped into families based on demographic attributes or conceptual relationships. It also helps to maintain a research journal, or diary, for the project in order to keep track of tasks performed, questions that come up, and required follow-up actions. This has the added advantage of functioning as an audit tool so that the researchers are later able to show their work and better illustrate their methodology (Lewins & Silver, 2007).

Retrieving Coded Data

Data can be retrieved based on coding trends and targeted keyword searches. It can also be meaningfully accessed via reflexive memo writings. Memos may be linked to individual codes or data segments, but they also may be linked to entire documents. Writing is the greatest key to managing data; it captures the thinking process that occurs in each session, and enables this thinking to evolve over time. Writing provides continuity and reflexiveness to the analytic process (Lewins & Silver, 2007).

Validity

There are a few types of validity that are most relevant to CAQDAS, especially when dealing with auto-coding applications like Leximancer. *Face validity* is concerned with how well the software algorithms work to search, parse, and analyze the data. The theory and construction of these algorithms should be based on standard practices and proven through rigorous research. Face validity examines whether automated software processes are able to generate a set of abstracted categories, based on criteria expected by the user. *Correlative validity* compares the output of the software to other methods that are already widely recognized as valid (Smith & Humphreys, 2006). One way to determine correlative validity is to have a domain expert manually analyze the data in parallel to the automated software analysis (Penn-Edwards, 2006).

Finally, *functional validity* is concerned with whether the software is realistically able to serve its advertised purpose: facilitate the learning and analysis of qualitative data (Smith & Humphreys, 2006). Can the computer application efficiently and effectively help users understand and interpret the emergent concepts or themes within a substantial corpus of text?

Reliability

There are two major types of reliability that should be considered when working with CAQDAS: *stability* and *reproducibility*. Stability has to do with analysis of the same data over multiple trials. If a particular content analysis method is able to repeatedly produce similar results with the same data set, then it is considered reliable (Smith & Humphreys, 2006).

Inter-coder reliability is closely related to stability. This is concerned with whether multiple coders are able to code the same set of documents in a consistent manner, using a common coding scheme. In order to achieve inter-coder reliability, coders must often become familiarized and trained on using the established coding scheme. They must be on the same page in regard to how each code is defined, and when exactly a particular code should be applied to text segments (Smith & Humphreys, 2006).

Reproducibility looks at whether common coding patterns can be found when dealing with similar types of material. It also is concerned with the reverse scenario: do data sets that are known to be quite different from each reveal different conceptual patterns? Thesaurus classifiers and concept maps can also be evaluated for reproducibility. One may examine whether the same thesaurus can be reused with other texts from the same domain. In the same way, one might determine if similar concept maps are produced when the software package is fed two related texts.

Outputs

All CAQDAS software packages have the ability to generate and output various reports, in a multitude of formats, including Word, Excel, and SPSS. Often, these reports consist of quantitative statistics about word frequencies, code usage within documents, and weighted co-occurrences of concepts or phrases. In addition, entire coded data segments can be exported so that researchers are able to analyze and reflect upon the text offline, outside of the software. This provides a different vantage point, which can help if researchers get stuck with their thinking or interpretation. Exporting data is also valuable if one wants to analyze the data using a different set of software tools.

General CAQDAS Usage Tips

Regardless of the software package used, there are some general best practices that should be considered when using CAQDAS to conduct qualitative research. First or all, it is important to remember that a project file is simply a container for one's work. Its main purpose is to help provide both dynamic access to the raw textual data, and enable abstracted sensemaking of the conceptual and thematic patterns that exist within this content. Secondly, CAQDAS is designed for flexibility, in terms of both methodology and task sequence. This means it is a good idea to make methodological decisions from the outset, and remain aware that there is no required order. Qualitative analysis involves creative exploration, iterative steps, and a considerable amount of multi-tasking with different tools or techniques. Finally, there is no single, proper, or best way to use CAQDAS, and one shouldn't let the structure or limitations of software dictate methodology. Allowing this could inhibit fuller interpretation and understanding of the data (Lewins & Silver, 2007).

A Comparison of Leximancer, MAXQDA, and Dedoose

Leximancer is quite a different content analysis tool than MAXQDA and Dedoose. Its primary purpose is to analyze large bodies of text that would be prohibitively resource intensive to manually code with the other two tools. It automatically generates a compelling visual mind map that can be manipulated in various ways to show more or less concepts and larger or smaller themes. It interacts with the list of themes and concepts in the right sidebar. This sidebar is where all the deeper human analysis takes place, exploring the auto-generated concepts, the relationships between the concepts, and the raw text excerpts they were derived from. Because of its nature, Leximancer only accepts textual documents (no multimedia files). It also doesn't have much in the way of manual coding options, except that the thesaurus can be pre-seeded with particular terms you are interested in. Leximancer is the most expensive option for students and it doesn't support mobile devices at all. However, it does work on Windows, Macs, and Linux.

MAXQDA is a mature product that is well balanced with many features and a well laid out interface, for the most part. The use of colors and icons is a little bit over the top, but it is fairly intuitive to learn. MAXQDA has a unique feature in which emoticons can be used as graphical codes. The organization of documents and codes is more flexible than any of the other offerings. MAXQDA has the broadest file type support of any of the three software packages. It allows images, audio files, and video files to be imported, as well as segmented and coded. Memoing is also quite robust and flexible in MAXQDA. The Plus version has a feature called MAXDictio which can create an automatic index or dictionary of terms used in each document, and tell you the frequencies of each. This allows for greater support of mixed methods research. MAXQDA is now compatible with both Windows and Macs. It also has a dedicated IOS application. MAXQDA is quite expensive for a standard license, but students can get a license for as cheap as $100.

Dedoose is a much younger product than MAXQDA. It is a cloud-based, SAAS application that runs on the Adobe Flash/Air platform. This means it can be run in a web browser or a standalone Air application, on both Windows and Mac machines. It also works on Android mobile devices, but not iPad or iPhone, since IOS doesn't support Adobe Flash. Dedoose has the cleanest looking user interface, but some of its interactions can be too modal and tedious (e.g. activation/deactivation of project objects or manual reordering of codes). The best part of Dedoose is its Quick Code widget, which allows you to apply multiple new and existing codes to excerpts in an extremely efficient manner. No mouse is even required. The memoing workflow is a bit clumsier than in MAXQDA, but it is equally as flexible in that memos can be attached to almost any project object. Dedoose has the widest variety of analysis tools and charts for exploring code application patterns. However, Dedoose really sets itself apart with its advanced security and access permission features. It is also the best option for team collaboration, since the whole project, including imported media files, is synced and stored in the cloud, accessible from anywhere. A training feature allows a research lead to ensure that all coders are consistently applying codes to document excerpts, thus increasing inter-rater reliability. The cost of Dedoose is extremely reasonable. It is only $10.95/month for an individual student, or even less for a larger group of users. They also only charge you during months you actually login to the application.

References

Leximancer Manual (Version 4). (2011). Retrieved December 28, 2013, from https://www.leximancer.com/site-media/lm/science/Leximancer_Manual_Version_4_0.pdf

Penn-Edwards, S. (2010). Computer aided phenomenography: The role of Leximancer computer software in phenomenographic investigation. The Qualitative Report.

Smith, A. E., & Humphreys, M. S. (2006). Evaluation of unsupervised semantic mapping of natural language with Leximancer concept mapping. Behavior Research Methods, 38(2), 262–279. doi:10.3758/BF03192778

Weber, R. P. (1990). Basic content analysis. Sage Publications, Inc.

	Lexicamancer	MAXQDA	Dedoose
Presentation of data	- Graphical mind map is the centerpiece of the data presentation. It can be manipulated to show more or less themes and concepts, and the words are interactive, working in tandem with the right-hand sidebar. - Concepts and themes (the equivalent to codes and code categories) are listed to the left. - Clicking on combinations of concepts reveals the raw excerpts that contain the co-occurrence of terms. - Because Lexicamancer auto-codes the data, the analysis is reversed from typical CAQDAS apps that primarily involve manual coding. You start by looking at the codes, and then dig into these codes in to explore and understand the underlying data. This is more of a top-down approach. - The raw text can sometimes be difficult to parse and read because it is all grouped together in one block of plain text	- The default layout shows the document and code hierarchies down the left side. The raw data (document content) is displayed in the middle. The right side shows any "retrieved segments" from a query. - Clicking on a document in the left-hand tree or one of the retrieved segments, on the right, changes the content that is viewed in the center Document Browser, jumping to the relevant excerpt in context. - The codes are shown with brackets, just to the left of the segments they are attached to. Clicking on one highlights the corresponding segment.	- Dedoose has a really nice Home Page dashboard which shows a summary of how many documents, codes, excerpts (segments), etc. are in the project. There is also a hierarchical code tree, a list of media (documents, audio files, etc.), a list of all the excerpts in the project, and an interactive codes cloud. Finally, there is a box where you can view various charts about how codes and descriptor variables relate to each other. - When a document is open, the code tree moves to the right of the screen. Additionally, there is a box that shows any currently selected excerpt and its attached codes - All of the segments are highlighted with automatically determined colors. They remain highlighted even when an excerpt is not actively selected. The actual code names can only be seen in popup boxes that appear when you mouse over a particular excerpt.
Exploring patterns of code application	- The graphical mind map and right-hand lists of concepts work in tandem with each other - You can view themes and a summary of each concept, as well as how important or relevant each theme is in the dataset (based on the level of connectivity of its concepts) - Concepts are viewed in ranked order based on frequency. Clicking on a particular concept shows you a list of related concept words, ranked based on how likely they are to co-occur with the selected concept. - To better understand how Lexicmancer is defining and describing a concept, you can look a the Thesaurus. It shows a list of words that are considered to be associated with the selected concept. - Finally, you can determine the pathways that connect two concepts together. This shows you the degrees of separation between two concepts.	- You can view the total frequency that codes are applied, as well as how these codes are distributed across each document in the project - You can view how different codes are applied based on various document variables (e.g. gender, age) - There are visual charts to see the actual locations that codes are applied within documents (i.e. the paragraph number) - You can also view a code matrix that shows how often various pairs of codes co-occur with each other - Some, but not all charts, allow you to retrieve the actual segments related to a particular data point in a chart - Multiple charts can be popped up at a time and they can be rearranged for cross-chart comparison purposes	- Dedoose has many similar analysis tools to MAXQDA, in terms of viewing code frequencies across documents and descriptor variables. - There are a number of additional ways to view the code relationships including tag clouds, binary code presence/absence in a document, and how often a particular coder used a code - All charts are dynamically interactive, allowing you to click on a data point and view the actual raw excerpts associated with that data point. This interactivity is more robust than MAXQDA. - Unfortunately, Dedoose is too modal, in that only one chart can be viewed at a time. This makes it difficult to compare charts to each other. After bringing up a list of excerpts related to a chart data point, the modal window can't be dragged around, and nothing underneath the modal can be interacted with. You must close the current modal first. This can become a bit tedious.
Importing data	- Accepted text file formats: .doc, .docx, .pdf, .html, .xhtml, .htm, .txt, .xml, .rtf, .tsv and .csv - There is no support for Excel spreadsheets (they must be converted first to .csv) - Loading documents can be tedious because the application is written in Java. This means that the standard file dialog is not used, and thus you are always forced to begin at the root of your user folder with no easy way to search for a file/folder. You simply must drill all the way down to where your documents are located. - Whole folders of documents can be imported at once which is helpful and saves time - Documents are externally pointed to on your machine (no copy is made into the Lexicamcer project)	- Accepted text file formats: .rtf, .rtfd, .doc, .docx, .pdf, .odt, .xls, .xlsx, and .txt. - Direct support of Excel spreadsheet files is a big plus, especially for importing survey data - Unlike Lexicamcer, HTML and XML files are not supported - Multiple documents can be imported at once. - You can easily drag and drop documents into MAXQDA from the file browser - All documents imported are copied into the project, so it will no be affected if the original file is moved or deleted.	- Accepted text file formats: .doc, .docx, .rtf, .txt, .htm, and .html - No PDF or spreadsheet file format support is a major drawback to using Dedoose - Multiple documents can be imported at once - Documents are copied into the cloud. Project will not be affected if the original, local files are moved or deleted.
Multimedia support (photos, audio, video)	- Because of the nature of Lexicamcer's textual analysis engine, only text documents can be imported.	- Accepts images, audio files, and video files - Image file formats: .jpg, .gif, .tif, .png - Audio file formats: .mp3, .wav, .caf - Video file formats: .mp4, .mov, .avi - Windows version accepts even more audio and video formats, based on the codecs installed on the computer system - It is annoying that .m4v files are not accepted even though they are simply .mp4 files that have a renamed extension. - Multimedia files are copied into a global MAXQDA folder, but they are not included in the project file itself.	- Accepts audio and video files - Image files cannot be imported - Audio file formats: .mp3, .wav, .m4a, and .wma - Video file formats: .mp4 only - Files are copied into the cloud. Because of this, Dedoose charges $0.05 per hour of audio uploaded and $0.25 per hour of video uploaded (each month). The first 1 hour of audio and 30 minutes of video are stored for free. - Video file compatibility is very limited. All videos must be converted into .mp4. - It is annoying that .m4v files are not accepted even though they are simply .mp4 files that have a renamed extension.
Outputs	- The visual concept map can be exported to an image file in a variety of formats and sizes - The thematic summary, concept hierarchy, and thesaurus can be exported to HTML and XML files - The concept list can be exported to an HTML or CSV spreadsheet file - Various queries (sets of excerpts) can be exported to HTML - Various pairs of concepts can be exported as matrixes into CSV spreadsheet files	- Document and code lists can be exported to Excel - Retrieved segments (queries) can be exported to HTML, Excel, or RTF. - All of the analysis tool charts and tables can be exported to HTML, Excel. Some can also be converted into images files (.png, .svg). - Memos, and their properties, can be exported to HTML and Excel - Individual memos can be exported to Rich Text files or directly printed - Any document in the Document Browser or set of retrieved segments can be instantly printed.	- List of documents and their properties can be exported to Excel - Filtered sets of excerpts (segments) can be exported to Excel, Word, and plain text. You can select which excerpt properties (e.g. code applications, excerpt creator, etc.) should be included in the exported document. You can also choose whether to include the full or a shortened version of each excerpt - Memos can be exported to Excel or Word. A number of options allow you to include counts or full information for any object (e.g. document, excerpt, descriptor, code) that is linked to the selected memos. - For some reason, individual memos cannot be exported from the memo details screen - Descriptors can be exported to Excel - Depending on the analysis chart or figure, export formats include Excel and PDF. Strangely, the code cloud can not be exported to an image file; only a PDF. - There is no direct print feature
Organizing documents	- Document organization is not very flexible - You must go back to the "Load Data" project settings. Then you can check the files and folders you want to include in the analysis. - Documents are inherently organized by their native file organization on your computer (since Lexicamcer points externally to the original files) - Unchecking or checking documents forces you to re-run the whole project in order to generate the concepts, thesaurus, etc. - There is no way to filter out particular documents, once all the data is loaded. All of the text is treated as a single corpus.	- Documents can be organized into Document Groups that go one level deep (these groups can't be nested) - You can create multiple sets of documents. These sets simply contain pointers to documents in the main list - Documents can be activated or deactivated as desired and filtered by various document variables	- Documents cannot be hierarchically organized into categories. They simply appear in a flat list - The document list can be filtered in numerous ways, including by user, date/time, media type, number of excerpts, and length. They can also be filtered by various descriptor variables. - The Data Selector allows you customize your data set you see to only include certain documents
Managing and grouping codes	- Concept words (codes) are automatically and dynamically organized into themes - There is not really a way to manually organize the concept codes	- Codes are all listed in the Code System box on the left-hand side of the interface. - They can be hierarchically organized into multiple levels by dragging one code into another. - Codes can optionally be sorted into alphabetical order - They can also be assigned descriptions and specific colors. Later, codes can be filtered by these colors - Like documents, codes can be added to one or more sets (as pointers). This means they don't get moved from their place in the main tree. - Codes can be imported from other MAXQDA projects, as well as spreadsheet files - Codes can be activated/deactivated manually or based on particular variables and colors	- Codes can be hierarchically organized into multiple levels using drag-and-drop - They can be optionally be sorted into alphabetical order - Codes can be imported from a spreadsheet file - Codes can also be assigned descriptions and weightings (min, max, and default) - Unfortunately, editing the codes requires that you switch the "Codes" box into edit mode. This seems like an unnecessary step. At least it stays in edit mode indefinitely, however, still allowing you to use the codes normally. - Manually recording codes is unnecessarily tedious. For some reason you can't drag and drop, but instead have to use up and down arrow buttons repeatedly. This can be frustrating and time-consuming. - Codes can be activated/deactivated via the Data Set module.
Selecting segments and creating codes	- Segments cannot by manually created in an ad-hoc manner - Codes can not be manually applied to segments - Normally, Lexicamancer creates a brand new dictionary and thesaurus that is emergently created from the text itself. However, a priori concepts can also be seeded before the project is run. For example, adding a "semantic lens" adds a predefined list of positive and negative terms to the dictionary. Lexicmancer will then search for frequencies and co-occurrences of these words when it runs.	- Segments can be highlighted with the mouse or the keyboard arrows - The Windows version has keyboard shortcuts for adding new codes, as well as the last code added to a segment. For some reason, the Mac version does not have any keyboard shortcuts. - Highlighted excerpts can be dragged onto a code or codes can be dragged onto an excerpt. This is handy to have this two-way interaction. - Recently selected codes appear in the Quick Code bar drop-down. This makes them more easy to access. - You can also add codes to a Favorites pane, if you want a shortlist of persistent codes that you use often. - In-vivo codes can be added, which name the code with the content of your highlighted segment. - Double-clicking an applied code in the margin of your document allows you to quickly add a comment about why you added that code, if you so desire. - Applied codes can easily be removed by right-clicking on the code name in the margin and selecting Delete. - The latest version of MAXQDA has a unique feature in which you can create labeled emoticon codes. There are many emoticons available for use. These "emoticodes" can help bring your codes to life and make them more instantly recognizable. - MAXQDA does code weighting but in a different way than Dedoose. The weightings are applied directly to an excerpt, as opposed to attaching weighting values to the code itself. The Dedoose method seems to be more preferable.	- Dedoose wins hands-down in this category. Their 'Quick Code' module allows you to highlight excerpts and quickly add multiple codes without even using a mouse. Simply select your text with the arrow keys (holding down SHIFT). Pressing SPACE brings up the Quick Code widget. You can immediately start typing to filter down the codes and search for the one you want. Then you simply hit RETURN once the one you want is highlighted. You can do this repeatedly if you need to add more codes because the search field stays in focus. This is all quite elegant and efficient. - If you need a new code that you haven't created yet, simply type the new code name in the search field of the Quick Code widget and hit Enter. - All of the codes you add to a segment appear in the "Selection Info" box on the right. - If you want to delete a code, Just click the "X" next to the applied code in the Selection Info box. - Code weightings can also be adjusted from the Selection Info box, if a code has been defined with weighting values. - The optional Upcode mode/feature is very convenient in that it adds all of the parent codes of a sub code that is applied to an excerpt.
Memoing	- Memoing is quite basic and not very robust in Lexicamcer - There is a "Logbook" where you can add retrieved segments and then attach a comment to each of these saved segments.	- You can attach memos to a document, an excerpt, or a code. You can also create "free" memos that aren't linked to any project object - These memos include a title, an editable creation date, an icon type, and a main body. - Codes can be attached directly to the memo - Memos can be written with rich text features, including bolding, italics, bullet points, font type, and more - Memos can be viewed in a list with various properties in the Memo Manager. In this way, they can easily be cycled through and reviewed. You can look at all code memos, all document memos, or memos for the currently opened document	- Memos can be linked to documents, excerpts, descriptors, or codes. - Memos include a title and a body - You can also create memos that are not linked to other objects in the project - Memos can also be manually organized into hierarchical folders
Merging and splitting codes	- Concept codes can not be merged or split in Lexicamcer	- Codes cannot be merged. However, coded segments attached to one code can be copied or moved to another code.	- Codes can be merged with each other, and all of the excerpts attached to the secondary code will be applied to the primary (new merged) code.
Auto-coding	- Lexicamcer, at its heart, is a textual auto-coding tool	- You can do a keyword query to find all segments that contain a particular word or phrase. Then, you can have MAXQDA automatically apply a certain code to these segments, at a sentence or paragraph scope. - The Plus version of MAXQDA has a feature called MAXDictio. This automatically indexes the words used in each text document. This can be used to compare the vocabulary used in various documents. It can also be used to determine basic quantitative statistics about the content, such as frequency that terms appear in a text.	- Dedoose does not have any auto-coding features
Inter-rater reliability	- Lexicamcer makes inter-rater reliability a mute point because it is based on machine algorithms - As long as the project is run in the same way, the results should be consistent	- MAXQDA has a tool called "Intercoder Agreement." It allows you to compare two instances of the same document that were coded by different coders. It looks at whether codes exist in each instance of the document, the frequency of each code, and the percentage of segment agreement. This lets you know how reliably the coding schema is being applied across a data set, when multiple people are involved in the coding process	- Dedoose has a "Training" tool to help you train coders and ensure inter-rater reliability. You setup a test by selecting a set of codes and existing excerpts that you, as the research leader have already coded. Then, the participants look at each excerpt and apply the codes they think are appropriate. Dedoose ultimately compares these code applications to your original coding of these excerpts, in order to statistically determine the level of agreement and reliability.
Security	- None. Anyone using the local computer can open, view, and edit a project	- None. Anyone using the local computer can open, view, and edit a project	- Users must login with a username and password, anytime they use Dedoose - Dedoose automatically logs you out after being idle for a certain amount of time - Each user can be given a multitude of granular access permissions on a per-project basis. - Users can be organized into various access groups
Team collaboration	- None	- Users are tracked as authors when they import documents or create codes, memos, segments, etc.	- Documents, codes, memos, excerpts, etc. have user attributes attached to them - Data Sets can be filtered by particular users - Dedoose is the most team-friendly software package because the projects and all associated media are synced and stored in the cloud. They can be accessed from any computer device - Robust user security options allow the project manager to set access permissions based on the role each user is playing on the project. This keeps certain users from accidentally destroying or modifying data or coding.
Desktop compatibility	- This is a Java-based application that opens in a web browser. Therefore, it works on Windows, Mac, and Linux, assuming that Java has been installed.	- Versions available for both Windows and Mac	- This is an Adobe Flash/Air application which means it works in a web browser, as well as any other platform (e.g. Windows, Mac) that supports Adobe Flash/Air.
Mobile device support	- No mobile support, as these devices don't have Java installed.	- Dedicated IOS app (for both iPad and iPhone) - No Android support	- No IOS support since Adobe Flash is not supported in IOS - Works on Android and Windows mobile platforms
Student pricing	- $1346 for a perpetual academic license (one desktop) - $673 for an annual license (one desktop)	- $99 for the standard product - $119 for the Plus version which includes MAXDictio, a tool that allows you to create an index of words used in texts, to determine frequencies and compare words used across documents. - Additional pricing tiers for server-based network and PC lab licenses or a portable flash drive license	- Offered as a SAAS cloud service - $10.95/month - 10% discount if you pay for 6 months at once - 15% discount if you pay for a whole year at once - You are only charged for months that you actually log into Dedoose - Additional discounts for larger of users