



SOFTWARE TOOL ARTICLE

REVISED WordCommentsAnalyzer: A windows software tool for qualitative research [version 2; peer review: 2 approved]

Ehsan Abdekhodaie ¹, Javad Hatami ¹, Hadi Bahrami Ehsan ¹, Reza Kormi-Nouri ²

¹Department of Psychology, University of Tehran, Tehran, Iran

²Center for Health and Medical Psychology, Örebro University, Örebro, Sweden

v2 First published: 03 May 2018, 7:536 (<https://doi.org/10.12688/f1000research.14819.1>)
 Latest published: 04 Sep 2018, 7:536 (<https://doi.org/10.12688/f1000research.14819.2>)

Abstract

There is a lack of free software that provides a professional and smooth experience in text editing and markup for qualitative data analysis. Word processing software like Microsoft Word provides a good editing experience, allowing the researcher to effortlessly add comments to text portions. However, organizing the keywords and categories in the comments can become a more difficult task when the amount of data increases. We present *WordCommentsAnalyzer*, a software tool that is written in C# using .NET Framework and OpenXml, which helps a qualitative researcher to organize codes when using Microsoft Word as the primary text markup software. *WordCommentsAnalyzer* provides an effective user interface to count codes, to organize codes in a code hierarchy, and to see various data extracts belonging to each code. It also offers basic visualization tools. We illustrate how to use this software by conducting a preliminary content analysis on Tweets with the #successful hashtag. We also demonstrate that the software has satisfactory performance on a large dataset of Iranian journals abstracts. We hope this open-source software will facilitate qualitative data analysis by researchers who are interested in using Word for this purpose.

Keywords

Computer assisted qualitative data analysis software, Microsoft Word, comments, coding, thematic analysis, code hierarchy tree

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	2
REVISED version 2 published 04 Sep 2018		✓ report
version 1 published 03 May 2018	✓ report	↑ ? report

- 1 **Ronggui Huang**, Fudan University, Shanghai, China
- 2 **Yazdan Mansourian** ^{id}, Charles Sturt University, Bathurst, Australia

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Javad Hatami (hatamijm@ut.ac.ir)

Author roles: **Abdekhodaie E:** Software, Writing – Original Draft Preparation; **Hatami J:** Methodology, Supervision; **Bahrami Ehsan H:** Methodology, Supervision; **Kormi-Nouri R:** Supervision

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2018 Abdekhodaie E *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Abdekhodaie E, Hatami J, Bahrami Ehsan H and Kormi-Nouri R. **WordCommentsAnalyzer: A windows software tool for qualitative research [version 2; peer review: 2 approved]** F1000Research 2018, 7:536 (<https://doi.org/10.12688/f1000research.14819.2>)

First published: 03 May 2018, 7:536 (<https://doi.org/10.12688/f1000research.14819.1>)

REVISED Amendments from Version 1

Updates in the software:

- Many tweaks in analysis code to improve the software performance on large datasets.
- We added visualization tools: Code Co-Occurrence Matrix and File Code Matrix.
- We added search and move features in the code hierarchy.
- We added capability to drag and drop codes from the code list or code hierarchy into other software.

Updates in the manuscript:

- We added a few citations to make the introduction more recent and more convincing.
- We amended new features illustration to Use case 1 (with two additional Figures)
- We amended a distinct use case for performance of software on a large dataset of Persian/Arabic text.
- We amended a Table and discussion about comparison of the WordCommentsAnalyzer capabilities with other QDA tools.
- We revised the conclusion so that it better summarizes the points made in the article.

See referee reports

Introduction

Commercial qualitative data analysis (QDA) software tools such as NVivo, MAXQDA and Atlas.ti seem to be the most popular in the qualitative research community^{1,2}, especially in health research. For example, a study found that 763 published articles in the Scopus database (between 1994 and 2013) used Atlas.ti and NVivo in their work, and that the majority of these studies were published in health sciences journals³. However, learning to use these complex software tools may be inconvenient for some researchers. In fact, research has shown that learners of complex qualitative tools often struggle with confusions, frustrations, and feelings of inadequacy⁴. Moreover, using complex QDA software may create a feeling for the researcher that they are forced to work within the software structures⁵. Besides, the purchase of commercial QDA software may not be affordable for some researchers. On the other hand, free or open-source solutions that are available often do not provide a smooth editing and markup experience (e.g., QDA Miner Lite does not support Persian and Arabic languages; CATMA and CAT⁶ are not fast due to their web-based nature). For these reasons, some researchers use professional word-processing programs for their qualitative research projects.

The use of Microsoft Word for QDA is commonly documented^{7,8}. Using Word comments provides a straightforward way to annotate specific portions of the text and attach keywords or categories (codes) to them. However, as the amount of data grows, organizing codes in Word comments becomes an exhausting task.

In this article, we present *WordCommentsAnalyzer*, a free, open-source tool that allows qualitative researchers to

automate organization of the qualitative codes through a fast and easy-to-learn graphical user interface (GUI) while coding the textual material using Microsoft Word as professional, familiar word-processing software.

Methods

Implementation

This software is written in C# programming language using .NET Framework 4.5.2. The software also makes use of OpenXml library to extract comments from Word documents. Recent versions of Word store documents in XML format. OpenXml provides an easy way to query comments from a document. To facilitate assigning multiple codes to a piece of text, we assume a simple convention: different codes are entered in a comment with line breaks between them (as the descendant paragraphs of the comment element). The software uses a relational model approach to store the extracted codes and uses language integrated queries to collect different text portions related to each code, to calculate the code frequencies and to sort the codes by frequency. The main visual interface of the program consists of three side-by-side panels (Figure 1). The left panel shows the codes in the comments with their counts, the middle one provides a code tree that the user can intuitively organize their codes in and the right panel shows the data extracts pertaining to each code. In the left panel, the code list can be filtered to find specific codes. The user can place codes in the code hierarchy simply by using drag-and-drop. The tree also allows for moving codes in the hierarchy if needed. The user can introduce a new parent code or a code that is of a higher level of abstraction. Additionally, the codes are changed or combined by being wrapped in new codes. The code hierarchy tree is saved as a tab-indented text file in the data folder (code-hierarchy.txt). The tree is auto-saved every minute and can also be manually saved by clicking Save button. The previous tree files are backed up in a subfolder of the data folder. When a collection of codes develops after coding several documents, the user can drag and drop the codes into the word-processing software to avoid memorizing them. In addition to organization tools available in the GUI, the software offers two visualization tools: Code Co-Occurrence Matrix visualizes the number of co-occurrences of sets of two codes in the data and File Code Matrix visualizes the number of occurrences of each code per Word document.

Operation

The requirements for this software are Windows 7 or later and .NET Framework 4.5.2. After installing the .NET Framework, the user can unzip the latest release package from the [GitHub link](#) and run the “WordCommentsAnalyzer.exe” executable file. The program supports XML Word documents (using the .docx extension). Older Word documents (using the .doc extension) can be easily converted to XML documents by Word 2003 or later (there are also resources available on the web to batch-convert older Word documents). The program allows multiple Word files to be analyzed. This feature can be utilized to separate transcripts of different interview or focus group sessions into different files.

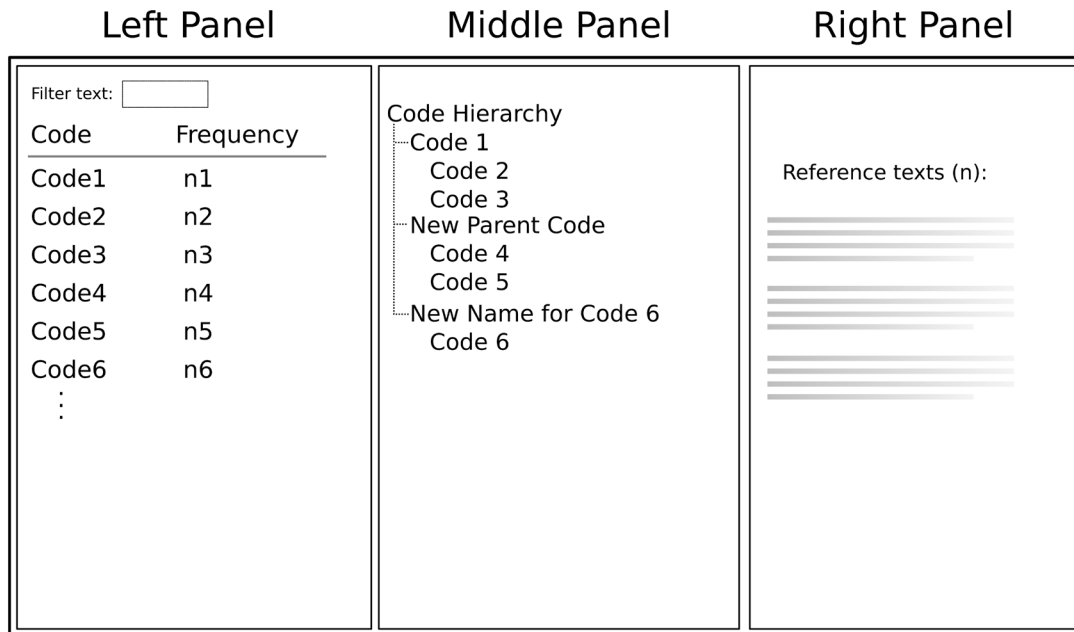


Figure 1. An illustration of the three side-by-side panels of the *WordCommentsAnalyzer* graphical user interface (GUI). The left panel shows the codes in the comments with their counts, the middle panel provides a code tree for intuitive organization of the codes and the right panel shows the data extracts pertaining to each code (or to children of a parent code). The code list in the left panel can be filtered to find specific codes. The user can place codes in the code hierarchy simply by using drag-and-drop. The tree also enables the user to move codes in the hierarchy if needed. The user can introduce a new parent code. The codes are changed or combined by being wrapped in new codes.

Use case 1

To illustrate how to use the software, we first present a mini-study of Twitter’s Tweets from 17 January 2017 to 10 April 2018. The Tweets with the #successfulaging hashtag were copied into two Word documents based on the year in which the Tweets were posted (Supplementary File 1). We reviewed the Tweets and added comments (line-break-separated codes) to portions of texts containing interesting notions related to successful aging. Two examples of these text portions are reproduced in Figure 2.

After adding comments to Word documents, we run *WordCommentsAnalyzer*, select the folder containing the Word documents and click *Analyze*. The program analyzes the comments and shows a list of codes with their counts in the left panel. The middle panel enables us to organize the codes by placing them in a code hierarchy (Figure 3). For example, we can find several codes related to health by filtering the code list by the word of “health”. Then we add the code of “Health”, which is a parent code, to the hierarchy by dragging and dropping it onto the root node (“Code Hierarchy”) or the empty area. The codes of “Brain health”, “Physical health”, and “Health care” can then be drag-and-dropped onto the node of “Health”. Likewise, “Oral health” is inserted into “Physical health”. When organizing the codes, we could check the right panel to assure the data extracts support the codes. Also, the codes inserted into the hierarchy will be highlighted in the code list to help keep track of the organized codes.

As the number of codes in the code hierarchy increases, moving or reorganizing codes becomes cumbersome, particularly when the user intends to move a code to another distant code or find specific codes in the deeper branches of the hierarchy. The software offers two features for smooth reorganization of codes: search specific words in the hierarchy and move codes through a pop-up window (Figure 4). Consider we want to review all the codes containing “retirement” in the Tweets data. We type a portion of this word (“retire”); by looping through the results (clicking Down), we realize that the “Retirement communities” is currently a child node of the “Communities”. Thinking that this node better suits the “Retirement” node, we can right-click it and select Move Retirement communities, then search for the “Retirement” node in the pop-up window, and move the “Retirement communities” to its appropriate place.

Figure 5 presents a formatted version of codehierarchy.txt (Supplementary File 2) when we organized the Tweet codes with at least two counts. As shown in this figure, the themes of health, retirement, happiness and being active represent the richest themes in the Tweets of #successful aging.

WordCommentsAnalyzer also allows for getting basic visual representation of the data. By clicking Visualize, a new window with two tabs appears. In Code Co-Occurrences Matrix tab, we see two identical instances of the codes lists. By checking codes in these lists, the software builds a co-occurrence matrix with the checked codes in the lists as the columns and rows.

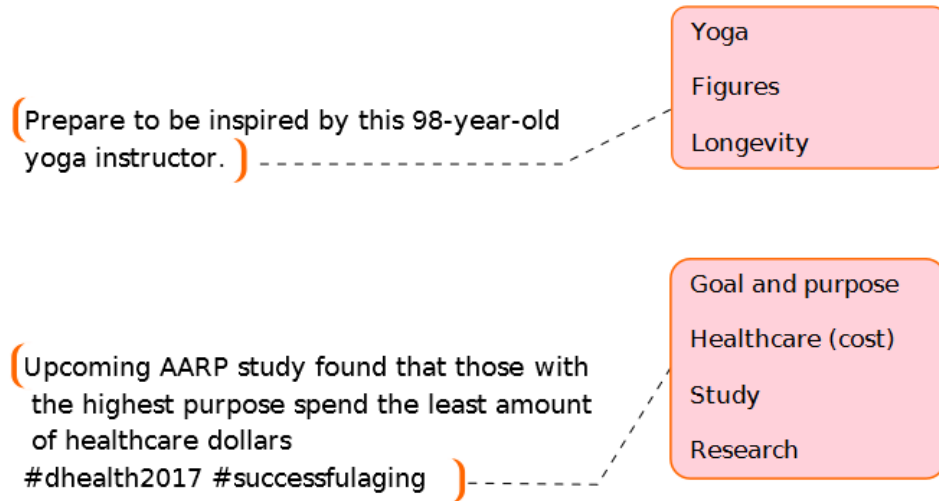


Figure 2. Two text samples of #successfulaging Tweets, which are commented using line-break-separated codes. The codes describe notable topics concerning the text samples.

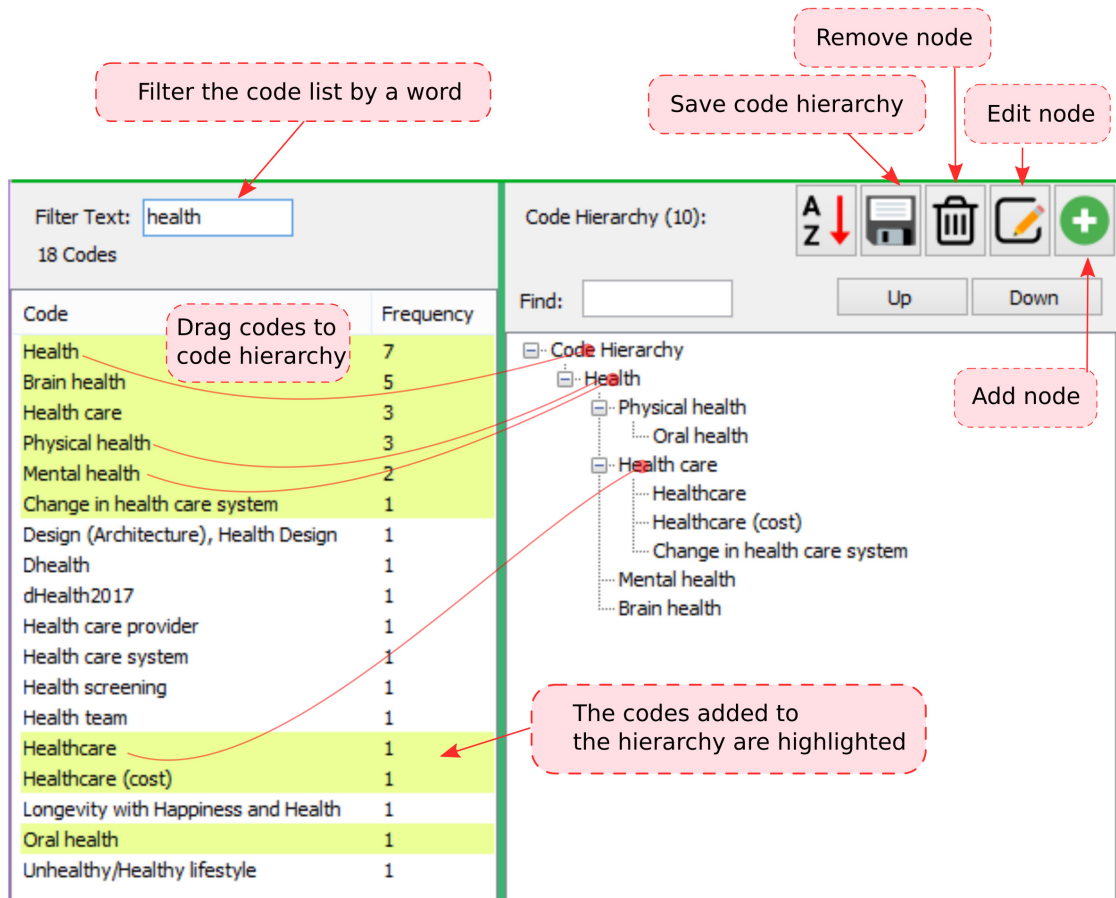


Figure 3. Basic features in the left and middle panels of the *WordCommentsAnalyzer* graphical user interface (GUI). The user can find specific codes by filtering the code list (e.g., by the word of "health") and organize the codes (from the left panel) by dragging and dropping them into the code hierarchy tree (the right panel).

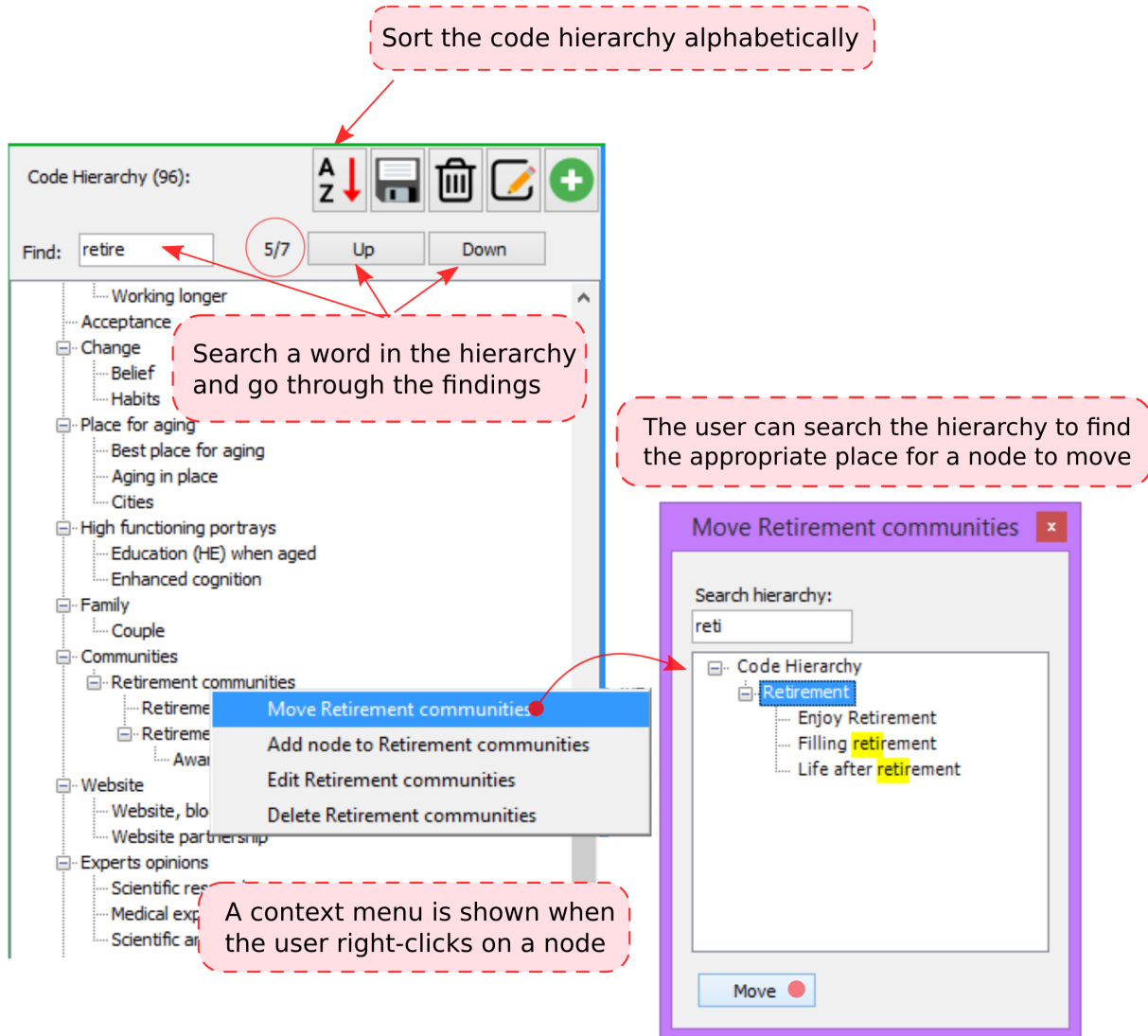


Figure 4. Search and move code features of WordCommentsAnalyzer which facilitates reorganization of codes in the code hierarchy. WordCommentsAnalyzer facilitates finding and moving specific codes by two features: 1) the user can search particular words in the hierarchy; 2) the user can move codes to other codes that are not visible in the current view by means of a pop-up window.

The numbers in the matrix cells show the number of text segments that share the corresponding pair of codes and the cells' color intensities are associated with the c-coefficients⁹. Creating the co-occurrence matrix for the Tweets data, allows inferring some thematic proximity between the codes with high co-occurrence. For instance, that the codes of "Longevity" and "Figures" have relatively high co-occurrence shows the Tweets' tendency to present model figures as very aged (see Figure 6a). Also, the high co-occurrence between "Marketing" and "How to" may indicate that the Tweeters often use "How to" phrases for marketing purposes.

In the File Code Matrix tab, we can generate a matrix of the number of paragraphs with a certain code in each document. For example, the matrix in Figure 6b demonstrates that while

the number of Tweets coded with "Longevity" was smaller than ones with "Brain health" in 2017, the former is greater than the latter in 2018.

Use case 2

The purpose of this use case was twofold: to test the performance of WordCommentsAnalyzer against a large dataset and to test the software when working with Persian/Arabic texts.

We collected the abstracts of eight Iranian journals in health sciences published until Aug 2018. We collected each journal issue into a Word document and assigned the keywords as codes for each abstract (All the journals were licensed by Creative Commons, CC BY 4.0 or CC BY-NC 4.0; the commented Word files and code hierarchy text file are available in

1	Health	51	Goals
2	Physical health	52	Planning ahead
3	Eating	53	Goals and dreams
4	Nutrition	54	Perception of aging
5	Fitness	55	Aging population
6	Health care	56	Role models
7	Mental health	57	Figures
8	Mental exercise	58	Models
9	Brain health	59	Active
10	Enhanced cognition	60	Physical activity
11	Mental exercise	61	Exercise
12	Homecare	62	Athletic
13	Longevity	63	Running
14	Longevity with Happiness and Health	64	Marathon
15	How to	65	Social engagement
16	Marketing	66	Social relationships
17	marketing	67	Friends
18	Marketing strategies	68	Social relationships (absence)
19	Technology	69	Social isolation
20	Critical	70	Loneliness
21	Contrarian	71	Social network
22	Biomedical model	72	Staying active
23	Social model of successful aging	73	Volunteering
24	Definition	74	Working longer
25	Retirement	75	Acceptance
26	Enjoy Retirement	76	Change
27	Filling retirement	77	Belief
28	Life after retirement	78	Habits
29	Retirement communities	79	Place for aging
30	Retirement community	80	Best place for aging
31	Retirement village	81	Aging in place
32	Award	82	Cities
33	Event	83	High functioning portrays
34	Expo	84	Education (HE) when aged
35	Challenge	85	Enhanced cognition
36	New experiences	86	Family
37	Happiness	87	Couple
38	Meaning	88	Communities
39	Optimism	89	Retirement communities
40	Never too late	90	Retirement community
41	Athletic	91	Retirement village
42	Record	92	Website
43	Worth	93	Website, blog
44	Life after being aged	94	Website partnership
45	Wisdom	95	Experts opinions
46	Growth	96	Scientific research
47	Aging diseases	97	Medical expert
48	Dementia	98	Scientific article
49	Caregiving	99	Dance
50	Learning	100	Pet

Figure 5. A formatted version of tab-indented text output file of the code hierarchy tree. When we organized the Tweet codes with at least two counts. The large branches of the code tree can help the researcher identify the richest themes in the data. Thus, themes of health, retirement, happiness, and being active are probably the major themes in the Tweets with the hashtag #successfulaging.

(a)

	Marketing	Longevity	Technology	Critical	Definition	Retirement	How to	Health	Event	Brain health	Challenge	Exercise	Figures	Happiness
Marketing	-	3	-	4	1	-	12	-	5	-	-	-	-	-
Longevity	3	-	-	-	-	-	3	9	-	-	5	6	14	6
Technology	-	-	-	-	-	-	-	-	3	-	-	-	5	-
Critical	4	-	-	-	17	-	-	-	-	-	-	-	-	-
Definition	1	-	-	17	-	-	-	-	-	-	-	-	-	-
Retirement	-	-	-	-	-	-	6	-	-	-	-	-	-	-
How to	12	3	-	-	-	6	-	3	-	-	-	-	-	3
Health	-	9	-	-	-	-	3	-	-	-	-	-	-	6
Event	5	-	3	-	-	-	-	-	-	-	-	-	-	-
Brain health	-	-	-	-	-	-	-	-	-	-	-	7	-	-
Challenge	-	5	-	-	-	-	-	-	-	-	-	-	-	-
Exercise	-	6	-	-	-	-	-	-	7	-	-	-	-	-
Figures	-	14	5	-	-	-	-	-	-	-	-	-	-	-
Happiness	-	6	-	-	-	-	3	6	-	-	-	-	-	-
Physical activity	-	-	-	-	-	-	-	-	-	-	2	5	-	-
Aging diseases	5	-	-	-	-	-	-	-	-	-	-	-	-	-

(b)

	Marketing	Longevity	Technology	Critical	Definition	Retirement	How to	Health	Event	Brain health
2017.docx	●	●	●	●	●	●	●	●	-	●
2018.docx	●	●	●	●	●	●	●	●	●	●

Figure 6. Basic visualization features of WordCommentsAnalyzer: Code Co-Occurrences Matrix (a) and File Code Matrix (b). WordCommentAnalyzer offers two visualization tools. Code Co-Occurrence Matrix enables the researcher to recognize patterns of codes co-occurrence in the data. The numbers in the co-occurrences matrix are counts of text segments that share the corresponding pair of codes and colors of the cells reflect c-coefficients⁹. High co-occurrence suggests thematic proximity between a pair of codes. For instance, in the co-occurrence matrix generated by the software for the Tweets data (a), there is a high co-occurrence between the “Longevity” and “Figures” codes, which shows that the Tweets tend to present model figures as very aged. File Code Matrix assists the researcher in inspecting the various data parts (e.g., different interview or focus group sessions) in terms of the codes or themes they contain. For example, this figure (b) demonstrates that while the coder(s) coded more Tweets with “Longevity” than they coded with “Brain health” in the 2017 document, this pattern was reversed in the 2018 document.

Supplementary File 3). The dataset was quite large, comprising 388 files, 4624 paragraphs, and 10378 codes. We tested the software on an ASUS U41J laptop (Intel(R) Core(TM) i5 CPU M480@2.67GHz processor; Hitachi SATA/300, 5400 RPM hard drive). The analysis was completed in a few seconds. We organized 1000 frequent codes into the code hierarchy. Although the number of the nodes was large, all the panels remained responsive and the search functions responded almost instantly. Also, the visualization tools had decent performance; creating The Code Co-Occurrence Matrix took a few seconds when the matrix was smaller than 100x100 and took less than 30 seconds when it was as large as 500x500; the software created File Code Matrix for 1000 codes and 388 files in about 30 seconds.

The software also performed well in all the operations including code search when working with Persian/Arabic characters.

Comparison of WordCommentsAnalyzer with other available tools

As mentioned at the introduction, WordCommentsAnalyzer is based on the idea that users code the textual data in the word-processing software and subsequently organize the open codes in an effective user-friendly environment. Thus the users of this software are not able to do analysis on non-textual data such as images, audio, and video. Besides, the users must do operations of coding, re-coding, and removing codes on the word-processing side. Therefore we compared only analytical

capabilities of the software for textual data (i.e., what is done after open coding) with other QDA software (Table 1). In contrast to other tools, WordCommentsAnalyzer provides no memo-writing features. It offers features to count codes, to do simple queries on the codes and to organize them in a hierarchy. However, it does not allow complex queries (e.g., using Boolean

or proximity operators). The recent version of the software generates basic visualizations such as Code Co-Occurrence Matrix and File Code Matrix but does not provide sophisticated visualizations like mind mapping tools. It is noteworthy that, RQDA (another free, open-source software) offers similar features as well but not in the GUI (the user have to write R syntax).

Table 1. Analytical capabilities of some well-known QDA software along with the WordCommentsAnalyzer capabilities.

The table presents analytical capabilities of three popular commercial QDA software tools and one free, open source QDA program along with features of WordCommentsAnalyzer. Because the focus of WordCommentsAnalyzer is on the analytic work after coding data, we did not include features of these software tools used for raw data manipulation and/or coding. Although WordCommentsAnalyzer offers no memo-writing or mind-mapping features, it provides features to do simple queries on the codes and to organize them in a hierarchy. In addition, it provides basic visualization tools.

MAXQDA 2018 (Commercial)	
Create memos and attach to the codes or data	<ul style="list-style-type: none"> The user can create free and attached memos (attached to documents, codes, etc.) The user can log notes about the research process in the logbook. Other writing tools included are paraphrases, comments, and summaries.
Search and interrogate the dataset	<ul style="list-style-type: none"> The user can see frequency of codes across the dataset. The software provides Boolean and proximity operators for retrieving coded text. The user can find co-occurrences of codes in up to six dimensions.
Visualize codes and the data	<ul style="list-style-type: none"> The user can get a big picture of tones of codes in a document (Document Portrait), see counts of codes per document (Code Matrix Browser) or counts of sets of two codes across the whole dataset (Code Relations Browser). The user can also see how paragraphs in a document are coded (Code Line) and compare this to other documents (Document Comparison Chart). The software provides MAXMaps which is a tool allowing visual mapping of all aspects of the project. The user can also add items not existent in the project to the map or create a 'free' mind map. MAXMaps provides a variety of automatic templates to generate visualizations, e.g., a map of codes hierarchy or a map showing code co-occurrences.
Nvivo 12 (Commercial)	
Create memos and attach to the codes or data	<ul style="list-style-type: none"> The user can create memos and link sources to them. The user can link a source to only one memo.
Search and interrogate the dataset	<ul style="list-style-type: none"> The user can see frequency of codes across the dataset. The software provides various types of queries to search text, find cases with a combination of codes, attributes and/or specific texts The queries can be saved for later use.
Visualize codes and the data	<ul style="list-style-type: none"> A variety of visualization tools including Mind Maps, Concept Maps, Project Maps, Matrix Query, and charts of cluster analyses of word similarity.
Atlas.ti 8 (Commercial)	
Create memos and attach to the codes or data	<ul style="list-style-type: none"> The user can create memos and link them to Quotations, codes and other memos. The user can also write Comments which are spaces granted specifically to different project components (e.g., Code Comments).
Search and interrogate the dataset	<ul style="list-style-type: none"> The user can search for words in the name, content or comments of the project components (e.g. codes, documents, memos, groups, networks, etc.), with an option to search on the basis of the user that created the components. The Query Tool provides Boolean and proximity operators for retrieving quotations based on the codes. Additionally, it provides 3 semantic operators (UP, DOWN, SIBLINGS) allowing retrieval based on the presence of transitive links between codes. The user can retrieve Quotations based on two-code co-occurrences in a tree view (Co-Occurrence Explorer) or a matrix (Co-Occurrence Table).
Visualize codes and the data	<ul style="list-style-type: none"> The user can create highly customized networks of the linked components. The user can arrange the components in a network using a number of automatic layouts offered. The software shows the relationships between codes in the Co-Occurrence Table. The table cells have shades of colors which indicate the strength of the relationship by c-coefficient.

RQDA 0.3-1 (Free, Open source)	
Create memos and attach to the codes or data	<ul style="list-style-type: none"> The user can create memos for codes, cases, and files but not for individual data extracts.
Search and interrogate the dataset	<ul style="list-style-type: none"> The user can search codes effectively by R commands but cannot search the codes for particular words in the GUI when they code data extracts, organize codes, etc.
Visualize codes and the data	<ul style="list-style-type: none"> The software can create a map of codes belonging to code categories and arrange them by a number of automatic layouts.
WordCommentsAnalyzer 2.0.3.0 (Free, Open source)	
Create memos and attach to the codes or data	<ul style="list-style-type: none"> The software does not provide the memo-writing feature.
Search and interrogate the dataset	<ul style="list-style-type: none"> The user can see frequency of codes across the dataset. The user can search the code list or code hierarchy for particular words when organizing the codes.
Visualize codes and the data	<ul style="list-style-type: none"> The software provides a matrix of code co-occurrences for selected codes. The table cells have shades of colors which indicate the strength of the relationship by c-coefficient. The software provides the File Code Matrix which visualizes codes in the Word files by a matrix of circles of varying size. The radius of circles inside the table cells represents the relative number of paragraphs coded by different codes in each file.

Note: the authors used software reviews by Silver, Lewins, and Bulloch¹⁰ and Silver and Bulloch¹¹ to create some parts of this table.

Conclusion

The rationale for developing WordCommentsAnalyzer was to facilitate organization and analysis of codes for researchers interested in using Word for data annotation. Despite that the query tools of this QDA software are somewhat limited and it includes no memo-writing/mind-mapping tools, it is free and open-source and provides basic code query and visualization tools through an easy-to-learn GUI. WordCommentsAnalyzer may provide a good option for researchers who see commercial QDA software as too advanced, complex or costly for their research purposes. By using this free software, the qualitative researcher can utilize a convenient word-processing application yet they reduce the efforts of manual organization of the codes.

Software availability

Source code available from: <https://github.com/ehsabd/word-comments-analyzer>.

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.1404728>¹².

License: GNU General Public License 3.0.

Grant information

The author(s) declared that no grants were involved in supporting this work.

Supplementary material

Supplementary File 1. Tweets hashtagged with #successfulaging from 17 January 2017 to 10 April 2018.

[Click here to access the data.](#)

Supplementary File 2. The tab-indented text file of code hierarchy.

[Click here to access the data.](#)

Supplementary File 3. Coded abstracts of eight Iranian journals in health sciences published until Aug 2018 along with code hierarchy text file.

[Click here to access the data.](#)

References

1. Lewis RB: **NVivo 2.0 and ATLAS.ti 5.0: A Comparative Review of Two Popular Qualitative Data-Analysis Programs.** *Field Methods.* 2004; **16**(4): 439–69.
[Publisher Full Text](#)
2. Hart T, Achterman P: **Qualitative Analysis Software.** In: Matthes J, Davis CS, Potter RF, editors. *The International Encyclopedia of Communication Research Methods.* Hoboken, NJ, USA: John Wiley & Sons, Inc. 2017.
[Publisher Full Text](#)
3. Woods M, Paulus T, Atkins DP, *et al.*: **Advancing qualitative research using qualitative data analysis software (QDAS)? Reviewing potential versus practice in published studies using ATLAS.ti and NVivo, 1994–2013.** *Soc Sci Comput Rev.* 2016; **34**(5): 597–617.
[Publisher Full Text](#)
4. Silver C, Rivers C: **Learning from the learners: The role of technology acceptance and adoption theories in understanding researchers' early experiences with CAQDAS packages.** *ATLAS.ti User Conference 2013: Fostering Dialog on Qualitative Methods, Technische Universität Berlin.* Berlin: Universitätsverlag der TU Berlin. 2014.
[Publisher Full Text](#)
5. Woods M, Macklin R, Lewis GK: **Researcher reflexivity: exploring the impacts of CAQDAS use.** *Int J Soc Res Methodol.* 2016; **19**(4): 385–403.
[Publisher Full Text](#)
6. Lu CJ, Shulman SW: **Rigor and flexibility in computer-based qualitative research: Introducing the Coding Analysis Toolkit.** *Int J Mult Res Approaches.* 2008; **2**(1): 105–17.
[Publisher Full Text](#)
7. Chenail RJ, Duffy M: **Utilizing Microsoft® Office to produce and present recursive frame analysis findings.** *Qual Rep.* 2011; **16**(1): 292.
[Reference Source](#)
8. La Pelle N: **Simplifying qualitative data analysis using general purpose software tools.** *Field Methods.* 2004; **16**(1): 85–108.
[Publisher Full Text](#)
9. Friese S: **Qualitative data analysis with ATLAS.ti.** London: Sage. 2014.
[Reference Source](#)
10. Silver C, Lewins A, Bulloch S: **Software Reviews: MAXQDA 2018.** CAQDAS Networking Project. 2018; 1–5.
11. Silver C, Bulloch S: **Software Reviews: ATLAS.ti 8.** CAQDAS Networking Project. 2018; 1–5.
[Reference Source](#)
12. Abdekhodaie E: **WordCommentsAnalyzer: A windows software tool for qualitative research (Version 2.0.3.1).** *Zenodo.* 2018.
<http://www.doi.org/10.5281/zenodo.1404728>

Open Peer Review

Current Peer Review Status:  

Version 2

Reviewer Report 12 September 2018

<https://doi.org/10.5256/f1000research.17614.r37967>

© 2018 Mansourian Y. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Yazdan Mansourian 

Charles Sturt University, Bathurst, NSW, Australia

I reviewed the new version. The amendments addressed my initial comments and the paper can be indexed now.

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 02 August 2018

<https://doi.org/10.5256/f1000research.16128.r36504>

© 2018 Mansourian Y. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Yazdan Mansourian 

Charles Sturt University, Bathurst, NSW, Australia

This article reports an overall description about developing a free and open source software designed for qualitative data analysis. This software assists researchers to organise and categorise the initial codes they create through data analysis process in various qualitative methods such as thematic analysis, ethnography and grounded theory. As this software is free and easily accessible, it will be useful for many researchers who do not have access to more sophisticated tools.

The authors highlighted the ease of use and free access of this tool as its main benefits and I do agree with them. Nonetheless, they did not compare the capabilities and performance of this software with one or two well-known tools in this field to explain its strengths and weaknesses. Obviously, the existing tools have several options for deeper level of analysis and also some facilities for analysis of non-text data such as image, audio and video. As a result, I think the authors should also remind the reader about the inevitable limitations of this product. I also recommend the following revisions in the article:

1. The article should be more informative in terms of distinctive features of this software comparing to similar tools in this context.
2. The first citation in the introduction section (Lewis, 2004) has been published 14 years ago while since then the area of qualitative data analysis (QDA) has been developed considerably. As a result, I recommend to include a few recent citations in this section to provide the reader with a more accurate image of the current trends and issues in this area.
3. The case presented in this article is based on a relatively small collection of data. Accordingly, we still do not know how effective the software will be in larger data sets. As sometimes, the huge volume of the data in large scale studies may reduce the performance of a software.
4. The authors mentioned popular qualitative data analysis tools such as QDA Miner Lite does not support Persian and Arabic languages. Nonetheless, the case presented in the article is English and we still do not know about the real performance of this new tool in other languages such as Persian and Arabic.
5. The conclusion is too brief at this stage and it should be more detailed and insightful. A good conclusion summaries the main points of the article and invites the reader to think further about the focal point presented in the discussions.

In general, this article is based on a creative idea but still requires some revisions and I hope the software presented here will be useful for researchers who use qualitative approach in their studies.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 29 Aug 2018

Ehsan Abdekhodaie, University of Tehran, Tehran, Iran

We thank Dr. Mansourian for his valuable comments about our manuscript/software. The comments surely helped us improve both the software and the manuscript.

Some aspects of the software have been updated at the time we present this revised manuscript. Indeed, apart from the tweaks done in the software performance so that it can handle larger datasets, comparison of the software features to the features of other well-known QDA software compelled us to add some useful features to our software including search in code hierarchy feature and visualization tools. We also specifically respond to the reviewers comments in the following:

1. We compared the features of this software to the similar tools in a new Table in the manuscript and added an accompanying paragraph which highlights the main points regarding this comparison and describes the specific limitations of our software. We also mentioned the limitations of WordCommentsAnalyzer regarding analysis of non-textual materials, doing complex queries, or presenting sophisticated visualizations.
2. We added a few citations. Specifically, one to provide more recent data on the usage of popular QDA software (Woods, Paulus, Atkins and Macklin, 2016) and others to give the readers a glance on current challenges in learning complex QDA software (Silver and Rivers, 2014; Woods, Macklin, and Lewis, 2016).
- 3, 4. The software indeed had issues with large dataset as the reviewer suspected. Consequently, we have updated the software so that it can handle larger datasets more efficiently. Then, we added another use case that is based on a collection of Iranian journal abstracts to test the software performance on a large dataset. This way we could also address the reviewer's comment about performance of the tool in the presence of Persian/Arabic texts. It is worthy to note that, we developed the feature of search in the hierarchy and move in the hierarchy since we realized that it is not feasible for large datasets to find the codes by scrolling the hierarchy tree or to move the codes by merely dragging and dropping.
5. We tried to make the conclusion more informative and more linked to the rationale of the study. Also, we revised it so as to present a better summary of the points made in the article.

Competing Interests: No competing interests were disclosed.

Reviewer Report 11 June 2018

<https://doi.org/10.5256/f1000research.16128.r34052>

© 2018 Huang R. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Ronggui Huang

Department of Sociology, Fudan University, Shanghai, China

Various QDA software provide similar functionalities in terms of coding operations and organization of codes, for instance, RQDA (<http://rqda.r-forge.r-project.org/>), WeftQDA (<https://www.pressure.to/qda/>), Py3QDA (<https://github.com/Ronggui/PyQDA/>), among others. A more systematic comparison of existing tools and *WordCommentsAnalyzer* will provide a clear picture on the relative advantages and disadvantages of the latter.

It seems that *WordCommentsAnalyzer* mainly organizes the codes and shows the related coded text segments, but does not support coding operations on-the-fly. The operations of coding, remove coding, and re-coding have to be conducted on the Word Processor side. It would be helpful to potential users to describe this point clearly.

Since the operation of coding must be done with Word Processor, it seems that coders have to remember and type the names of codes directly or via the copy-and-paste method. It would be valuable if coders can do the coding via *WordCommentsAnalyzer*.

The most obvious advantage of *WordCommentsAnalyzer* is its easy use, especially for Windows users.

Overall, *WordCommentsAnalyzer* is a valuable new tool for organizing codes based on Word Processor.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Sociology

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 29 Aug 2018

Ehsan Abdekhodaie, University of Tehran, Tehran, Iran

We would like to thank Dr. Huang for reviewing our software/manuscript and for his kind comments about the potential value this software provides. Here we respond to the reviewer's comments as follows:

We added a 'more systematic comparison' of existing tools and WordCommentsAnalyzer through a table which included three commercial QDA tools and RQDA along with WordCommentsAnalyzer. We also discussed the relative pros and cons of the program.

We asserted that our program is not meant to do coding operations on-the-fly and these operations should be done on the word-processing side.

The reviewer mentioned the concern that it may be difficult for the researcher to memorize and type the names of codes or copy-and-paste them into Word. We added a feature that enables the user to drag and drop codes (from either the code list or the hierarchy tree) into the Word comment. So the user can both remain on the word-processing side for editing the files and reuse the developed code list/hierarchy effectively.

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research