SCALING Up Content Analysis: Crowd-Coding Text Units

Nicholas Adams
University of California, Berkeley
Berkeley, CA, USA
nickbadams@berkeley.edu

Abstract
This paper describes a new Crowd Content Analysis Assembly Line and enabling software allowing researchers to complete large-scale content analysis projects with the help of citizen scientists and/or crowd workers. By decomposing content analysis into cognitively simpler tasks that scale to a larger workforce, the approach can reduce the duration of large-scale content analysis projects by a factor of six or more.

Author Keywords
Text analysis; content analysis; qualitative; quantitative; human centered data science; natural language processing; HCI; CSCW.

ACM Classification Keywords
H.5.2. Natural language; H.5.3. Computer-supported cooperative work.

Introduction
Terabytes of textual data are being digitized and made available every day. But until now, scientists working with large quantities of text have had to face a dilemma: they can either analyze their large text corpora using automated methods that are often inadequate for theory-testing, or they can perform a richer analysis on a smaller set of data by painstakingly
categorizing and extracting variable/attribute data from the text by hand. So far, implementing this latter, ‘hand-coding’ text analysis approach, often called ‘content analysis,’ has been rather infeasible at the scale of thousands or more documents.

**The Challenge of Content Analysis**

The basic task of content analysis seems simple enough. A human reads through text looking for information specified in a researcher-created coding/semantic scheme that defines all the sorts of information – all the variables and attributes – of interest to the researcher. Each piece of information specified in the scheme has its own color and/or code, and the person performing content analysis must apply that code/color to the text using a highlighter and/or notation system. While this is performed on computers these days, the task is fundamentally the same task that – decades ago – was performed with highlighters, marginalia notation, and post-it notes.

The work is not particularly hard if the coding scheme is very simple. But, as the number of variables in the scheme grows, the requirement that coders compare the text to many variables as they read combines with the task of applying hand-codes in a way that can overwhelm the cognitive capacity of the sharpest mind.

Every hand-coder decision, every choice of a highlighter color (code), every application of the highlighting to text by a movement of the mouse pointer, uses limited cognitive resources. When hand-coders have to hunt through a deep and complex 100 variable coding scheme to choose a code, they can literally lose sight of the text to which it should correspond. They can blend their comprehension of a sentence with a nearby sentence as they attempt to hold multiple code application processes in their minds at once. These and other opportunities for simple cognitive errors abound. Even when researchers are excellent at hand-coding text, more codes (i.e. more variables) force them to process more information simultaneously, tiring their brains faster, forcing them to slow down or quit a hand-coding session sooner.

The Dynamics of Collective Action (DCA) project, led by McAdam, McCarthy, Olzak, and Soule provides a contemporary illustration of another key limitation of large-scale content analysis projects: they often rely on an unstable workforce of undergraduates who need intensive training [6]. The DCA team required the combined effort of hundreds of undergraduates, a dozen graduate students and six professors. All told, they spent over a decade gathering information about 22 variables describing 24,000 social movement events reported in the New York Times from 1960 to 1995. The scale of that human effort is staggering. Yet, no one who has uttered the words ‘big data’ would call this dataset ‘big.’ If scientists hope to find hidden patterns in the terabytes of digital text now available – patterns invisible to automated text analysis tools – they need to find a new approach to manual content analysis that scales.

**Decomposing Content Analysis into Crowd Work**

The rise of crowdsourcing presents researchers with an opportunity to recruit a new workforce into their efforts. Though the demographic and personality composition of ‘the crowd’ varies a bit from both the undergraduate and general populations, studies show that their work output generally does not [8]. However,
it is not immediately obvious how content analysis – work that is so cognitively taxing and requires so much training – could ever be organized into the sort of brief, well-defined tasks typically performed by crowds. In fact, some of the few researchers who have performed very simple and limited content analysis using crowds believe "it is impossible to write clear and precise instructions, to be understood reliably by a diverse, globally distributed set of workers in the crowd, for using a detailed and complex 56-category scheme quintessentially designed for highly trained experts [1].

This paper shows, on the contrary, that it is possible to reorganize content analysis work for schemes of 56 categories, 156 categories, or more. But, that reorganization relies relatively little on writing better instructions. It relies on a process that breaks the work out into sequential steps, none of which overload human cognitive capacity.

The problem must be approached from the standpoint of human cognition. Two features of a content analysis job can increase its difficulty: the length of texts, and the number of categories/variables/attributes in a coding scheme. The approach described here ensures that no single worker faces long texts with complex coding schemes.

Recall that a coding scheme lists in hierarchical fashion all the categories of information researchers want to identify in, or extract from, a body of texts. The highest level of a hierarchical coding scheme will specify the researchers’ units of analysis — the human individual, the firm, the event, etc. One hierarchical step down, variables will describe those units of analysis. 'Hair color' and 'height' are variables, for instance, that might describe a ‘human individual’ unit of analysis (but not a firm or an event). And, at the lowest level, attributes like 'brown,' 'black,' 'red,' and 'blonde' will describe variables like 'hair color' (but not height).

Since attributes only pertain to some variables, and variables only pertain to some units of analysis, some branches of a coding scheme (units of analysis and variables), and all of those branches’ leaves (attributes) will be entirely irrelevant to a given bit of text. When a person is reading text describing a firm, for instance, she will have no use for the branch of the coding scheme designed to capture information about hair color and height.

This hierarchical, branching structure of a coding scheme introduces an opportunity to decompose content analysis projects. One can think of a large, complex coding scheme as a collection of smaller coding schemes, each pertaining to a separate unit of analysis.

For crowds (who do brief tasks without face-to-face training) to effectively and efficiently help with content analysis, they can only be expected to apply a small coding scheme to a small unit of text that contains information relevant only to that coding scheme. They need to work with text units that correspond to a single unit of analysis at a time. We call these Text Units of/for Analysis (TUAs).

The TUA is the key that unlocks crowd content analysis. When a crowd worker reads a TUA, she is reading a few sentences, or fewer – a reasonable length of text for a reading comprehension task. And, that text is packed with information relevant to a single branch of a
The Possibilities

The possibilities for this crowd content analysis approach extend as far as the availability of text data and the imaginations of researchers. Some researchers will be interested in legal documents, others in policy documents and speeches. Some may have less interest in a particular class of documents and more interest in units of text ranging across them—perhaps related to the construction and reproduction of gender, class, or ethnic categories.

TextThresher may also serve as model for scaling the human analysis of other forms of qualitative data. Other researchers may wish to apply this approach to qualify and classify image, audio, and video data. ML algorithms from such approaches can then feed into solutions for major, long-standing AI challenges around qualitative data.

researcher’s coding scheme, requiring her to do little or no distinguishing between relevant and irrelevant information.

Of course, TUAs must first be identified in text. This is a task that can be performed by a traditional research team or crowd workers. Here, the text – perhaps a full news article – is rather long. But, the coding scheme is simple, with only as many categories as there are units of analysis under study (often 5 or fewer).

Software in Progress

At the time of this writing, a web interface designed to reorganize slow-going content analysis based upon the above task decomposition strategy is under development. This TextThresher software guides users through two steps. First, trained research assistants or crowd workers identify TUAs (text units corresponding to a single case of a single unit of analysis) in larger documents. Next, a second interface will display those TUAs to crowd workers and walk them through a series of leading questions about the text.

By answering these (usually) multiple-choice, reading comprehension-style questions, and highlighting the words justifying their answers, crowd workers extract detailed variable/attribute information relevant to the researcher’s semantic scheme while labeling the text that corresponds to those variables/attributes. Thus, the crowd completes work equivalent to traditional content analysis.

To recapitulate, this content analysis work is achievable as crowd work because researchers reduce text units from document length to a few sentences, because those few sentences are only relevant to a small branch of the larger semantic scheme, and because so many people are familiar with reading-comprehension tasks.

More Than Just Content Analysis

This crowd content analysis approach will be improved and extended in a number of ways that advance human-centered data science more generally. TextThresher is currently designed to use both “gold standard” data to automatically pre-train and pre-test crowd workers’ proficiency with a task [5], [10], and “voting,” a method in which crowdsourcers acquire redundant worker output, for instance from five crowd workers, and only store data about which four or more crowd workers agree [5], [2], [4], [14], [16].

The crowd content analysis process can be accelerated, too, with “active” machine learning [11], [19], [13], [17], [3]. Crowd-labeled text can be used to train models that classify text as humans have. The output from such models, predicting the variables/attributes of words in previously unlabeled text, can then be included in the crowd worker voting system described above, so that labeling may be completed with fewer crowd worker judgments [12].

To improve efficiency further, researchers may also build models (as demonstrated by [12], [18], [21], [9], [15], [7], [20]) matching particular extraction tasks (e.g. about firm TUAs but not individual TUAs) to those citizen scientists who have shown highest proficiency in faithfully completing tasks of that category.

References


The Possibilities

The possibilities for this crowd content analysis approach extend as far as the availability of text data and the imaginations of researchers. Some researchers will be interested in legal documents, others in policy documents and speeches. Some may have less interest in a particular class of documents and more interest in units of text ranging across them—perhaps related to the construction and reproduction of gender, class, or ethnic categories.

TextThresher may also serve as model for scaling the human analysis of other forms of qualitative data. Other researchers may wish to apply this approach to qualify and classify image, audio, and video data. ML algorithms from such approaches can then feed into solutions for major, long-standing AI challenges around qualitative data.

researcher’s coding scheme, requiring her to do little or no distinguishing between relevant and irrelevant information.

Of course, TUAs must first be identified in text. This is a task that can be performed by a traditional research team or crowd workers. Here, the text – perhaps a full news article – is rather long. But, the coding scheme is simple, with only as many categories as there are units of analysis under study (often 5 or fewer).

Software in Progress

At the time of this writing, a web interface designed to reorganize slow-going content analysis based upon the above task decomposition strategy is under development. This TextThresher software guides users through two steps. First, trained research assistants or crowd workers identify TUAs (text units corresponding to a single case of a single unit of analysis) in larger documents. Next, a second interface will display those TUAs to crowd workers and walk them through a series of leading questions about the text.

By answering these (usually) multiple-choice, reading comprehension-style questions, and highlighting the words justifying their answers, crowd workers extract detailed variable/attribute information relevant to the researcher’s semantic scheme while labeling the text that corresponds to those variables/attributes. Thus, the crowd completes work equivalent to traditional content analysis.

To recapitulate, this content analysis work is achievable as crowd work because researchers reduce text units from document length to a few sentences, because those few sentences are only relevant to a small branch of the larger semantic scheme, and because so many people are familiar with reading-comprehension tasks.

More Than Just Content Analysis

This crowd content analysis approach will be improved and extended in a number of ways that advance human-centered data science more generally. TextThresher is currently designed to use both “gold standard” data to automatically pre-train and pre-test crowd workers’ proficiency with a task [5], [10], and “voting,” a method in which crowdsourcers acquire redundant worker output, for instance from five crowd workers, and only store data about which four or more crowd workers agree [5], [2], [4], [14], [16].

The crowd content analysis process can be accelerated, too, with “active” machine learning [11], [19], [13], [17], [3]. Crowd-labeled text can be used to train models that classify text as humans have. The output from such models, predicting the variables/attributes of words in previously unlabeled text, can then be included in the crowd worker voting system described above, so that labeling may be completed with fewer crowd worker judgments [12].

To improve efficiency further, researchers may also build models (as demonstrated by [12], [18], [21], [9], [15], [7], [20]) matching particular extraction tasks (e.g. about firm TUAs but not individual TUAs) to those citizen scientists who have shown highest proficiency in faithfully completing tasks of that category.

References

Analysis: Reliable and Agile Production of Political Data." American Political Science Review (forthcoming).


