

# Textual Data Analysis Software Matures

## SPSS Text Analysis for Surveys

By Frank LoPresti  
frank.lopresti@nyu.edu

An interesting question I faced when reviewing SPSS Text Analysis for Surveys was whether I was looking at the emergence of a hot new technology or at something everyone else had already seen. Language pattern recognition software has, after all, been around for a while.

This software processes text and enables the use of powerful computers and large databases to categorize textual data. Researchers in the Social Sciences often need this capacity when asking open-ended questions (rather than, for instance, multiple-choice questions) of large numbers of people. The use of open-ended questions is important to many studies, and researchers often want to include the answers to these questions in their findings in more than an anecdotal fashion. Categorizing textual responses with tools like SPSS Text Analysis allows researchers to include this data in their quantitative analyses.

Imagine a scenario in which a dot-com company asks me to conduct a client satisfaction survey. I distribute a survey to 12,000 of the company's clients, asking multiple-choice questions such as "How long have you been a customer?" The clients answer by filling in the appropriate bubble: "Less than one year," "Two to four years," etc. In addition, my survey has several open-ended

questions such as "What do you like or not like about our service?" that the clients respond to with a few sentences.

**All told, SPSS Text Analysis for Surveys is an impressive tool, greatly facilitating the process of including textual data in quantitative analysis.**

When the time comes to analyze the results of my survey, I can use my scanner software<sup>1</sup> to read the multiple-choice answers into a dataset for quantitative analysis. I can then easily organize this data into tables, showing, for example, that most of the dot-com's clients have used the company for less than a year.

But what do I do with the text I collected in response to the open-ended questions? By simply reading the written answers, I could probably get a general idea of what the clients are happy and unhappy with, but what I really want to know is if there are common themes to their answers. If I can find common themes, I can create tables using these themes as categorical variables and then

include these answers in my report. The problem, then, is how to extract the themes in the first place.

### THE HISTORY OF TEXT ANALYSIS

In the old days, researchers had to manually evaluate surveys for themes, then quantify each response based on those themes. For example, in our sample scenario, if I established "friendly customer service" as a theme, I would have looked at each of the 12,000 responses, assigning "true" if the theme was discussed in some manner and "false" if it was not. I'd then repeat this laborious process for each theme I could identify.

For years, this was how text was converted to categorical variables for analysis, which explains why open-ended questions were not popular in large surveys. Even during the first two decades of university computing, the situation did not improve because memory was scarce and expensive, and computational linguistics had to use punched cards for input. As faster, more powerful computers and more versatile external media (e.g., floppy disks) developed, character data became easier to input, store, and manipulate. The text analysis programs that emerged were, however, still limited by small hard drives. They could merely parse single words and index

1. See [http://www.nyu.edu/its/pubs/connect/archives/fall02/lopresti\\_question.pdf](http://www.nyu.edu/its/pubs/connect/archives/fall02/lopresti_question.pdf).

or categorize text by whether it contained a single word and synonyms to that word.

About ten years ago, as computer memory constraints were reduced, software was developed that could parse characters into words, count them, locate specific words and their synonyms and, with somewhat limited success, the opposite of a word. There were, however, obvious challenges that still had to be faced, such as how to deal with misspellings and verb conjugations. If only computers could think!

As gigabytes have lead to terabytes and massive amounts of text have been stored digitally (consider the Internet), language analysis has taken giant steps. Statistical, fuzzy analysis of language has evolved into software that *does* seem to be thinking, addressing issues like “more important” versus “less important” words, “negative” versus “positive” usage, constructs across languages, idioms, phrases, and complex patterns.

At Princeton University, WordNet<sup>®</sup> was developed by the Cognitive Science Laboratory under the direction of Professor George A. Miller (Principal Investigator). To quote their website, “WordNet is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives, and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets.”<sup>2</sup>

You’ll be tested on that later, but essentially, linguists have been busy developing software that allows computers to look at terabyte collections of text and address previously mentioned challenges, such as misspellings. Large dictionaries, self-learning programs that extract concepts and link them into synonymous sets, and open source archives that index word relationships, such as WordNet, have been

created. Norman Nie, co-founder of SPSS and a professor at Stanford and the University of Chicago, built upon research such as this to create SPSS Text Analysis, a product that calls upon a MySQL database, WordNet, LibTextCat<sup>3</sup> (a classification library used for language guessing), and other software.

### SPSS TEXT ANALYSIS IN ACTION

Now, let’s walk through the text analysis of our fictitious scenario discussed earlier. First, I create a file with two variables: the ID of the respondent and the respondent’s text answer to the question “What do you like or not like about our service?” I import this file (which can be in Excel, SPSS, or ODBC format) into SPSS Text Analysis and run an EXTRACT.

To quote the SPSS help, “...text responses must be run through an extractor engine before categorization can begin. Using powerful linguistic resources, the engine identifies relevant concepts in your data and extracts them. The result of this extraction is a set of terms, types, and patterns that you can then categorize manually or automatically using built-in classification techniques.”

The software’s extractor uses its libraries on the responses to the question and, within a few minutes, determines three sets (lists) of words and concepts, which it calls “terms,” “types,” and “patterns.” The software creates the following list of types: “Positive,” “Negative,” “Locations,” “Names,” “Company Clients,” and “Products.” The software makes examination of the text very easy; I can open any type to see the phrases it includes. If I click on a type, all the responses with that type are separated out of the total responses and the suspected “Negative” text is highlighted. I am impressed, for example, to see that “so so slow” is typed as “Negative.”

The next step is to review the extracted sets and clean them up. For example, looking at the “Names” type, I see the names of staff but also “Victoria’s Secret.” Ha, ha... our dot-com had Victoria’s Secret as a web client. So, I clean up the list by moving Victoria’s Secret to “Company Clients.” I also specify that any instances of “Victoria’s” + something should be considered synonyms. That way, if one usage of “Victoria’s” points to the type “Company Clients” and all usages of “Victoria’s” are synonyms, when I run EXTRACT again, all references to “Victoria’s” are correctly classified. As I clean, I periodically rerun EXTRACT.

SPSS Text Analysis relies on a set of dictionaries created and manipulated by WordNet and LibTextCat, but any refinements I make as I clean the data are saved in a project dictionary that takes precedence over the other dictionaries. If I wish, I can use this custom dictionary and the categories I’ve established on other survey sets, making future analyses run much faster.

When I am satisfied with the sets of extracted terms, types, and patterns, I move on to the last step of building categorical variables for these responses. I can build categories from any combination of the extracted terms, types, and patterns. For example, if I display the set of terms by frequency, “E-mail” and its synonyms are at the top of the list with 5,700 hits. Since I certainly want to use it as a categorical variable, I do so by dragging it to the “Create Categories” window. The dataset I am building now has a variable called “E-mail” with 5,700 “true” values associated to the 5,700 people who discussed e-mail in their responses. If I drag and drop “Negative” into a second category, I have a second variable pointing at negative text.

When I export this session, I have an SPSS dataset with three variables:

2. <http://wordnet.princeton.edu/>

3. <http://software.wise-guys.nl/libtextcat/>

*Continued on p. 21 >>*