MWSUG 2025 – Paper VR-034

From Word Clouds to Phrase Clouds to Amaze Clouds: A Data-Driven Python Programming Solution To Building Configurable Taxonomies That Standardize, Categorize, and Visualize Phrase Frequency

Troy Martin Hughes

ABSTRACT

Word clouds visualize the relative frequencies of words in some body of text, such as a website, white paper, blog, or book. They are useful in identifying contextual focus and keywords; however, word clouds—as commonly defined and implemented—suffer numerous limitations. First, multi-word phrases such as "data set" or "Base SAS" are unfortunately segmented into single words—"data," "set," "Base," and "SAS." Second, desired capitalization often cannot be specified, such as visualizing "PROC PRINT" even when its lowercase "proc print" is observed in text or code. Third, spelling variations (e.g., single and plural nouns, various verb conjugations, abbreviations and acronyms) are not mapped to each other. Similarly, and fourth, comparable words or phrases (e.g., "PROC PRINT" and "PRINT procedure") are not mapped to each other, representing a further lack of entity resolution. This text and its Python Pandas solution seek to overcome data quality, data integrity, and data standardization issues that plague word clouds, by defining and applying configurable taxonomies—data models that can impart more meaning and precision to ultimate word/phrase cloud visualizations. The result is a phrase cloud that amazes—an amaze cloud!

INTRODUCTION TO WORD CLOUDS

Word clouds visualize the relative frequencies of words in some body of text—*or at least they aim to do so* while suffering a variety of afflictions. Some of these issues are discussed previously and can be partially or fully overcome, whereas others cannot or are not addressed in this text. Despite the progress that can be made toward "more accurate" or "more desirable" word clouds, including a shift from word clouds to phrase clouds, few would claim that word clouds represent a tremendously exacting visualization tool. Thus, although word clouds can charm or even convey some useful information, they exist for many as little more than a curiosity, and for the wary, evoke more disdain than 3D pie charts!

Throughout this paper, the text of my 2024 book is used as the substrate for analysis: *PROC FCMP User-Defined Functions: An Introduction to the SAS® Function Compiler*. Myriad online solutions facilitate the import of raw text to create word clouds, and www.wordclouds.com is used herein for all examples.

As a first step, the MS Word document of the book's manuscript is analyzed to display its contents, which reveals a book that nearly topped 100,000 words, as shown in Figure 1. Not bad for a book that was only contracted to be 120 pages—somebody sure is long-winded!

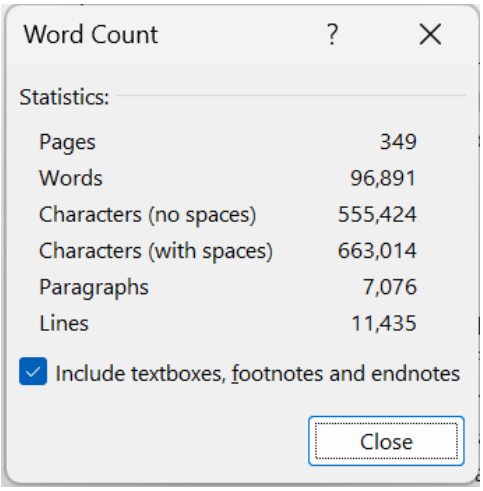

Figure 1. Page, Word, and Character Count of PROC FCMP Book

Next, the entire text can be copied and pasted into www.wordclouds.com, and in mere seconds, a word frequency table is created that displays the top 2,500 words, as shown in Figure 2.

Figure 2. Weighted Word Frequency of PROC FCMP Book (www.wordclouds.com)

But wait, this is a weighted frequency count showing *relative*—not *actual*—word frequencies! For example, "function" appears in the manuscript 1,372 times, not 1,426. And even without this comparison, several issues are immediately apparent from only this smattering of the dozen most common words:

- In converting all words to lowercase, original case has been lost, so "sas" is shown in lieu of "SAS."
- "function" is accompanied by "functions," so would it be appropriate to bin these together? But wait, both "function" and "functions" can function as a verb or a noun!! Thus, if verbs are being binned together, should not "functioned" and "functioning" be added to the lot?
- "can" is ubiquitous as a helping verb, but functions as a noun, so the analysis has included it as a relevant word. Perhaps in the tuna or cat food industries "can" would have some significance, but in the SAS world, it does not.

- "data" is super popular, but as this analysis represents only *word* frequencies, this subjugates all instances of "data set" as well as "DATA step" merely to the more generic "data."
- Similarly, "step" as a single word is confusingly elevated—until you remember that SAS practitioners are constantly discussing our beloved "DATA step."

Notwithstanding these salient issues, users still flock to word clouds because of the simplicity with which they can be generated. Taking only seconds to parse an entire textbook to generate a relative frequency count, www.wordcoouds.com immediately renders the corresponding word cloud, shown in Figure 3.

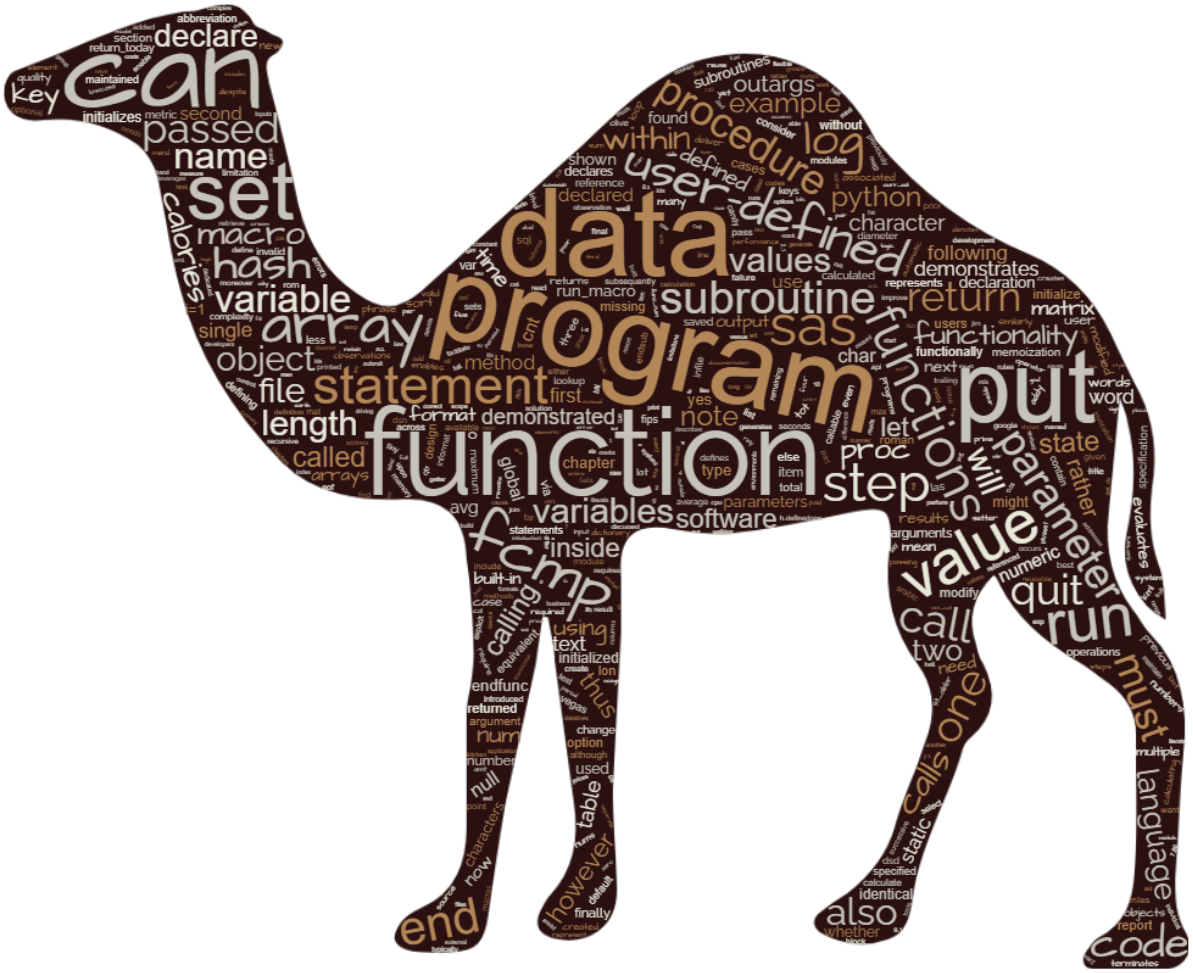

Figure 3. Raw Word Cloud Without Modification

For some purposes, this word cloud might suffice. But do you want "can" appearing on the camel's face? Do you want "must" tramp-stamped on his meaty thigh, or "also" on the camel's toe? I don't like my camels musty, so in the following sections, the data model that analyzes word frequency is progressively refined and elaborated to create more useful and contextually accurate phrase clouds.

ANALYZING MULTI-WORD PHRASE FREQUENCY

Consider the following phrase that will be parsed for evaluation throughout this text:

> "PROC FCMP, the SAS function compiler, empowers SAS practitioners to build user-defined functions that can be called by the DATA step to transform data."

And it do! Feeling empowered yet? FCMP allegiance aside, at the heart of word clouds are the underlying text parsing algorithms, which often exist as somewhat of a black box to end users. At a high level, some

parsing of punctuation and/or capitalization is required to standardize the data, after which words are evaluated singly and aggregated into a word frequency table.

For example, a traditional word parser would first detect "proc"; then "fcmp"; then "the"; and so forth, until the entire text had been evaluated. This would yield the following most commonly occurring words:

- the – 2
- sas – 2
- to – 2
- data – 2

To construct a *phrase* cloud, on the other hand, multi-word phrases must be identified in text, and this requires words to be evaluated not individually but with respect to their neighbors. Thus, a traditional word parser would fail to recognize "PROC FCMP" as a distinct entity (or the "SAS function compiler" as an equivalent entity), or the "DATA step" as a unique entity; this is unfortunate.

The solution is to calculate a phrase frequency table, which includes not only individual words but also multi-word phrases, with phrase length limited by user configuration (such as no longer than four-word phrases). The Python solution builds such a phrase frequency table by first segmenting a body of text into anecdotal "sentences"—groupings of words separated by punctuation tokens, such as commas, semicolons, periods, parentheses, and line breaks. This approach redefines the preceding text into three "sentences":

- PROC FCMP
- the SAS function compiler
- empowers SAS practitioners to build user-defined functions that can be called by the DATA step to transform data

Thus, in parsing the first two (of three) "sentences," two-word phrases are evaluated first—some will be meaningful, and others will not:

- PROC FCMP
- the SAS
- SAS function
- function compiler

The frequency of each of these multi-word phrases is then added to the word frequency table, so that it includes both single words and two-word phrases.

Next, in this iterative process, three-word phrases are evaluated. The first "sentence" ("PROC FCMP") is shorter than three words, so no phrases are identified. Subsequently, the second and third "sentences" are evaluated, and the following three-word phrases are identified—as before, some are meaningful while others are not:

- the SAS function
- SAS function compiler
- empowers SAS practitioners
- SAS practitioners to
- practitioners to build
- to build user-defined [and so on until all "sentences" have been evaluated]

The Python solution produces a list of tuples, each of which identifies a word or phrase followed by its frequency. Switching back from this simplified example to the PROC FCMP manuscript, the following

phrases contain the word "data" in order of frequency—once again, irrespective of whether a phrase connotes any useful meaning:

- ('data', 1309)
- ('data step', 461)
- ('the data', 382)
- ('the data step', 286)
- ('data set', 281)
- ('a data', 86)
- ('data _null_', 83)
- ('data step in', 74)
- ('data step in program', 72)
- ('the data step in', 66)
- ('the data set', 59)
- ('a data step', 54)
- ('from the data', 47)
- ('data sets', 46)
- ('the data set work', 39)
- ('data set work', 39)
- ('data steps', 34)
- ('in the data', 34)
- ('data structure', 33)

Notice that these frequencies are not mutually exclusive, so "data" appears 1,309 times in the text, including all subsequent usages listed, such as "DATA step" and "data set." Also note that common phrases might not be useful, such as "the data" or "a data." Thus, at the same time that phrase analysis can add value and understanding, by supporting more complex or exact meaning, the majority of "phrases"—that is, any sequence of words—will be meaningless or not add value.

In sum, where phrases are analyzed, it is insufficient merely to select the most common phrases—and just as commercial word clouds typically remove common words like "the" or "a" that add no value to a visualization, phrase analysis, too, must support some method to separate the hemp from the chaff. Artificial intelligence (AI) or machine learning models could be employed for this task; however, this solution employs the somewhat tedious albeit more precise construction of a taxonomy to identify valuable phrases.

BUILDING A CONFIGURABLE TAXONOMY

The preceding list of phrase frequency tuples represents only a handful of the more than 27,000 one-, two-, three-, and four-word phrases identified in the PROC FCMP book. That is, it would be infeasible to evaluate all phrases to determine if each has meaning. The solution, however, becomes clear when three of the phrase frequencies are inspected:

- ('the data', 382)
- ('the data step', 286)
- ('the data step in', 66)

Each of these phrases begins with "the," and therein lies the solution—to identify words like "the" that should *never* be the first word in any phrase because it provides no meaning. Thus, by adding a business rule that phrases starting with "the" should be excluded from analysis, far fewer phrases must be evaluated for meaning. Similarly, would a phrase ending with "the" provide any additional meaning than the preceding words alone? No, so a second business rule should dictate that phrases ending with "the" similarly are excluded from analysis.

Thus, the Python solution proceeds with the evaluation of all words as well as two-, three-, and four-word phrases, and constructs a raw frequency "table"—maintained as a Python list of tuples. The ten-most common words or phrases follow, and expectedly, are single words:

1. ('the', 6702)
2. ('to', 2396)
3. ('and', 2256)
4. ('a', 1921)
5. ('is', 1602)
6. ('in', 1547)
7. ('of', 1413)
8. ('function', 1372)
9. ('data', 1309)
10. ('that', 1242)

Next, for each of these most common words or phrases, the Python solution prompts the user to answer four questions:

1. Does this word or phrase provide meaning on its own?
2. Can this word or phrase start a meaningful phrase?
3. Can this word or phrase be in the middle of a meaningful phrase?
4. Can this word or phrase terminate a meaningful phrase?

The user would likely indicate that "the" has no meaning on its own and that it can neither start nor end a meaningful phrase. And because the user indicates this, when the Python solution identifies that there are 457 instances of the phrase "to the" in the manuscript, it will not query the user whether this is meaningful—because a rule has already been specified that "the" can neither start nor end a meaningful phrase. Thus, the Python solution adapts to past user responses to filter future questions about phrases that it poses to the user, and in so doing, quickly winnows out meaningless phrases. These user responses—the answers to the four questions for each word or phrase—are captured and maintained in a control table that can be utilized to evaluate not only this but also subsequent bodies of text. This data-driven programming approach maximizes software configurability, modularity, interoperability, and flexibility because the control table can be reused extensively—by different users, evaluating different bodies of text, and even by different programs and/or languages leveraging the same control table.

Finally, each word or phrase for which the user has indicated "Yes, this has meaning on its own" is added to the refined frequency table from which a refined phrase cloud is constructed.

ANALYZING MULTI-WORD PHRASE FREQUENCY USING A TAXONOMY

With the preceding taxonomy built, including user identification of meaningful words and phrases, the full text is parsed a second time using the taxonomy. To illustrate, the three sample "sentences" are again parsed:

- PROC FCMP
- the SAS function compiler

- empowers SAS practitioners to build user-defined functions that can be called by the DATA step to transform data

The Python solution parses the following words and phrases in order, with only "meaningful" words and phrases added to the ultimate phrase frequency table; also note that phrases are evaluated in order of word length with the longest phrases checked first:

1. PROC FCMP – evaluated to be meaningful, per user entry in the taxonomy control table; thus, because these two words comprise a phrase, neither "PROC" nor "FCMP" are evaluated in isolation, nor are these two words added to the final frequency table.
2. the SAS function compiler – skipped because no phrase can begin with "the," per the taxonomy control table.
3. the SAS function - skipped because no phrase can begin with "the," per the taxonomy control table.
4. the SAS - skipped because no phrase can begin with "the," per the taxonomy control table.
5. the – skipped because the word was identified as being meaningless on its own, per the taxonomy control table.
6. SAS function compiler - evaluated to be meaningful, per user entry in the taxonomy control table; thus, no further words or phrases in this "sentence" are evaluated because all words have been exhausted.
7. empowers... – all phrases starting with this word are skipped, per user entry in the taxonomy that "empowers" cannot start a meaningful phrase.
8. SAS practitioners to build – skipped as not meaningful.
9. SAS practitioners to – skipped because no phrase can terminate with "to," per the taxonomy control table.
10. SAS practitioners - evaluated to be meaningful, per user entry in the taxonomy control table; thus, parsing continues with the word after "practitioners."
11. to... – all phrases starting with this word are skipped, per user entry in the taxonomy that "to" cannot start a meaningful phrase.
12. build user-defined functions that – skipped because no phrase can terminate with "that," per the taxonomy control table.

This parsing continues iteratively through all "sentences" in the entire text until all meaningful words and phrases have been identified. In evaluating longer phrases first, this method ensures that a phrase like "data set" is always evaluated before "data" (the first word), so that only "data set" is added to the phrase frequency table.

Thus, the application of the phrase taxonomy to text evaluation can be visualized by highlighting user-specified meaningful words and phrases in bold brackets:

- **[PROC FCMP]**
- the **[SAS function compiler]**
- empowers **[SAS practitioners]** to build **[user-defined functions]** that can be called by the **[DATA step]** to transform **[data]**

Note again that what constitutes and does not constitute "meaningful" is driven solely by the configurable taxonomy. Thus, in the revised frequency analysis, "data" would appear only once (not twice) because the second incidence of "data" would be attributed to the phrase "DATA step" rather than the single word "data."

Switching back to the text of the PROC FCMP manuscript, the following words and phrases are most commonly observed among those identified as meaningful (within the taxonomy control table); note that the tilde in "data~set" represents a space in this usage:

1. ('function', 1359)

2. ('program', 952)
3. ('array', 634)
4. ('functions', 544)
5. ('statement', 464
6. ('data~step', 461)
7. ('user-defined', 450)
8. ('subroutine', 409)
9. ('value', 406)
10. ('fcmp', 399)
11. ('length', 329)
12. ('data', 324)

So, "function" was identified previously as occurring in the manuscript 1,372 times—so why will only 1,359 occurrences appear in the phrase cloud? Well, because the difference (i.e., the "missing" 13 instances of "function") are usages that appear in separate, meaningful phrases (e.g., "function call").

This revised phrase frequency table is next uploaded to www.wordclouds.com and ingested; that is, the website is versatile enough to be able to analyze both full text or frequency tables. Figure 4 demonstrates the weighted frequencies of words and phrases in the program.

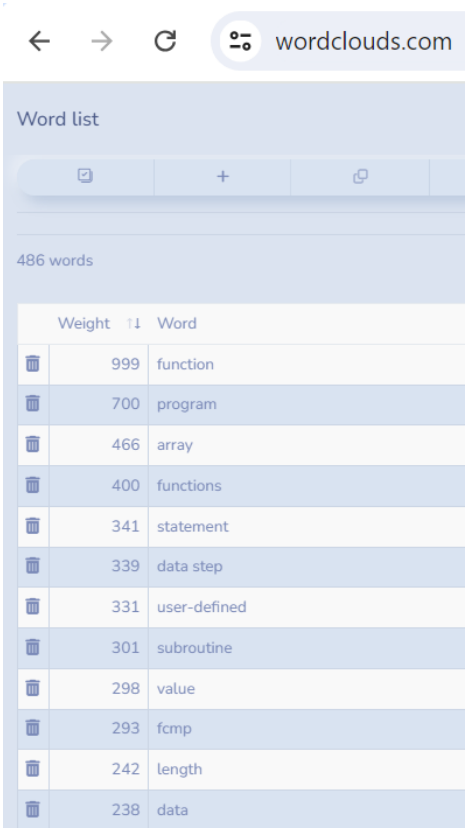

Figure 4. Weighted Phrase Frequency of PROC FCMP Book (www.wordclouds.com)

Thus, as compared with Figure 2, the words and phrases to be included in the phrase cloud now include phrases; additionally, all words and phrases have been identified as being meaningful in the configurable taxonomy. The result is a more refined, more useful phrase cloud, which is demonstrated in Figure 5.

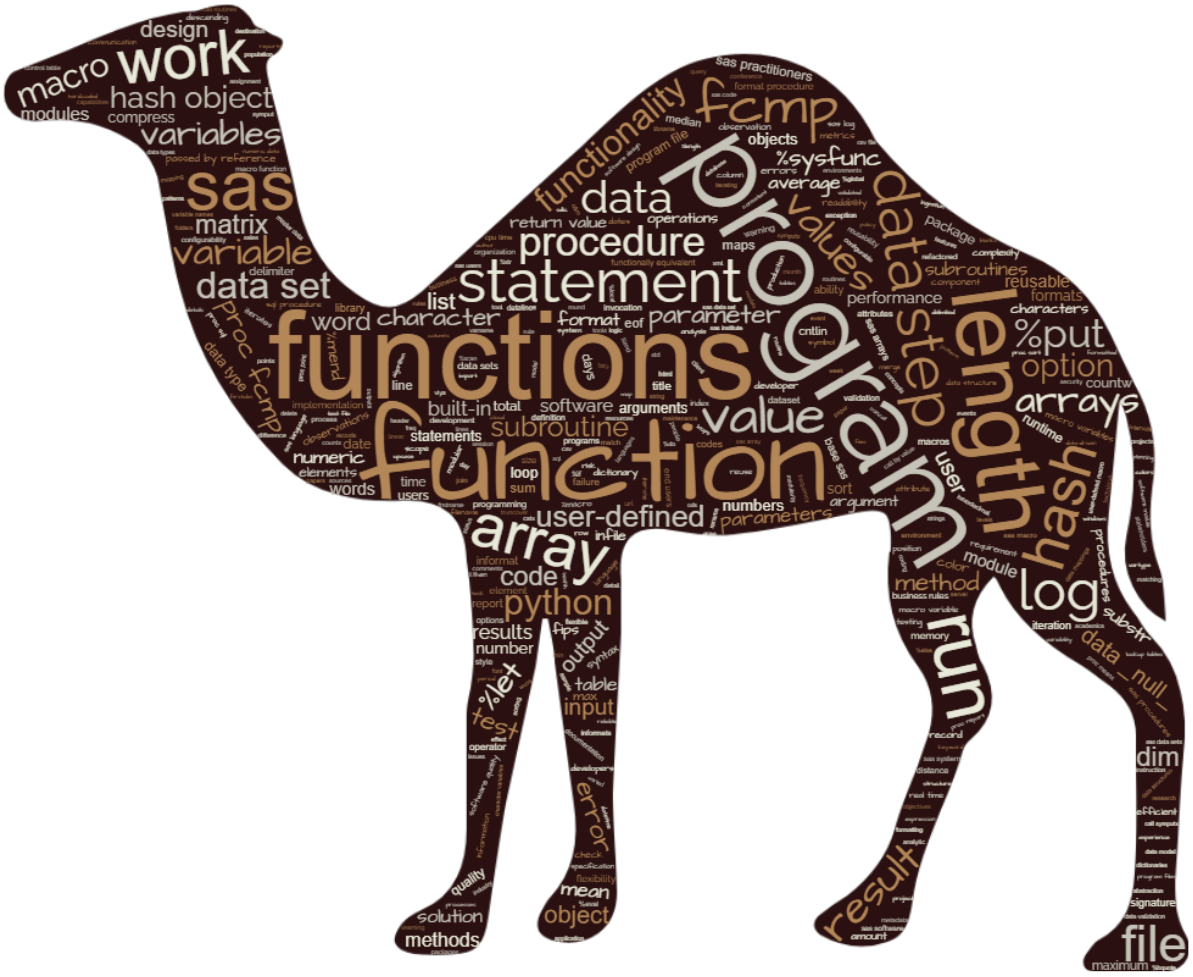

Figure 5. Refined Phrase Cloud with Taxonomy Applied During Text Parsing

Note that the camel in Figure 5, as opposed to the raw word cloud displayed in Figure 3, has overcome a couple issues:

- The phrase cloud is no longer just a word cloud, and includes phrases such as "data set" and "data step."
- The phrase cloud no longer includes meaningless words, such as "can" and "also" that had previously been included.

Although the phrase cloud is inarguably improved from the previous word cloud, it can be improved further. For example, note the issues that remain:

- Because text analysis is case insensitive, text case is not retained in the phrase cloud; thus, "SAS" is rendered as "sas"; "DATA step" is rendered as "data step"; and the macro statement "%PUT" is rendered as "%put."
- No entity resolution occurs, so both "function" and "functions" appear in the word cloud; this can diminish meaning or utility if words or phrases that essentially mean the same thing are repeated.

Thus, although a great first step in phrase cloud analysis requires the identification of meaningful phrases, a subsequent step can entity resolve those words and phrases—effectively binning and then aggregating terms to consolidate and eliminate redundancy.

BUILDING AN ENTITY RESOLUTION MAP

Entity resolution, in its simplest form, is a many-to-one mapping that transforms (or maps) any of a set a values to a single value. For example, as discussed in the previous section, it might be worthwhile to map all instances of "functions" to "function"; in Python a dictionary facilitates this, with key entries for both "function" and "functions" mapped to the value "function":

```
entity_dict = {'function': 'function', 'functions': function', '%put': '%PUT'}
```

Data-driven design prescribes that this dictionary should be maintained outside the program within a control table, after which the control table can be ingested, transformed into the preceding dictionary, and applied to the phrase frequency table to consolidate and aggregate entries. Thus, when the preceding dictionary is applied (after parsing of the text using the taxonomy but before generation of the phrase cloud), all instances of "functions" are changed to "function" and all instances of "%put" are changed to "%PUT." This increases the occurrences of "function" from 1,359 (as previously shown) to 1,865:

1. ('function', 1865)
2. ('program', 996)
3. ('array', 819)
4. ('value', 719)
5. ('PROC FCMP', 594)
6. ('statement', 563)
7. ('subroutine', 553)
8. ('variable', 544)
9. ('DATA step', 461)
10. ('user-defined', 448)

For a final time, the contents of the PROC FCMP book are ingested by www.wordclouds.com, but this time, after entity resolution. The resultant weighted phrase frequencies are shown in Figure 6.

Figure 6. Weighted Phrase Frequency of PROC FCMP Book (www.wordclouds.com) after Entity Resolution

And, for a final time, the resultant phrase cloud is shown, in Figure 7.

Figure 7. Entity-Resolved Phrase Cloud with Taxonomy Applied During Phrase Parsing

Now, as promised, Figure 7 reveals that "PROC FCMP" is indeed capitalized, as it should be, as well as other words and phrases that require or distinguish capitalization. Moreover, phrases such as "PROC FCMP" and "data set" are discernable.

So, is the juice worth the squeeze? For some, yes; for many others, no. Consider the metrics and the resources required to achieve the final phrase cloud:

- The taxonomy table of words and phrases contains 2,100 entries; each entry contains the answers of four questions and required a few seconds for the end user to complete.
- The entity resolution table that maps raw values to refined values contains 355 entries; some entries took seconds, such as mapping instances of "%put" to "%PUT" whereas others might have required more consideration as they are more existential. For example, should "SAS practitioner" be mapped to "developer" or are these two separate constructs or ideas?

Thus, although some issues with word clouds and phrase clouds have been overcome, enumerable, undeniable issues remain. For example, consider the random placement of words and phrases, the words and phrases that may not fit in a graphic, or which might not be viewable due to less-than-desirable font type, style, size, color, or other attributes.

UNRESOLVED ISSUES, JUST LIKE YOUR CHILDHOOD

Lest this subtle sense of accomplishment sink in, let's upend nearly all progress that has been made. Consider the word "data" as it appears in the PROC FCMP book; that is, the full word only, excluding hyphenations (e.g., "data-driven") and partial inclusions (e.g., "database"):

- "data" appears 1,309 times in the text, and thus 1,309 times in a raw word cloud without any refinement.
- After the taxonomy is applied, "data" appears only 324 times—because phrases containing "data" are deemed to be meaningful, such as "DATA step" and "data set." This diminution might more accurately capture the many facets of "data" while at the same time eclipsing the higher-level focus on "data."
- After entity resolution is applied, "data" still appears 324 times—because no other terms were mapped to "data," a subjective decision made entirely by the end user.

So, in this example, the decision about the relative size of "data" is fairly subjective, and defining a ton of phrases that include the word "data" actually diminishes the size of this word in the ultimate phrase cloud. Is it better to have one big "data" or sixteen smaller "data" representations?

Conversely, consider a second example. the frequency of the phrase "data set" in the PROC FCMP book:

- "data set" appears 281 times in the text; however, it would not appear in a *word* cloud because the phrase comprises two words.
- After the taxonomy is applied, which identifies "data set" as a meaningful phrase, the term appears 270 times in the text; the decrease occurs because "SAS data set" appears 11 times in the text, so this is subtracted from the 281 during taxonomy application.
- After entity resolution is applied, "data set" is represented as appearing 361 times in the text. This substantial increase occurs because multiple terms (e.g., "data sets," "SAS data set," "SAS data sets") are all mapped to "data set."

Inarguably, a phrase cloud that represents the SAS language is better served by a SAS-specific construct like "data set" and a taxonomy that similarly reflects SAS-specific nomenclature. How to represent and categorize this nomenclature, however, is largely directed (and constrained) by personal preference.

CONCLUSION

Word clouds have scantly been considered technical or accurate visualization tools, nor relied upon for serious decision-making. But where they are employed for curiosity's sake or the casual conveyance of information, they might as well be displayed as articulately and professionally as possible. First, definitely employ a camel. Second, consider identifying not only words but also phrases that are meaningful, as well as excluding those that are not. Third, consider mapping words and/or phrases through entity resolution. Fourth, review and revise the final product until that splanchnic sensation confirms the phrase cloud delivers the message you are intending to convey. Fifth, and most important, because you've employed reusable, data-driven programming principles, apply the same technology and taxonomy to subsequent, unrelated projects with ease!

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Troy Martin Hughes
E-mail: troymartinhughes@gmail.com