1 Simple search

1.1 Simple search inputs

1. Search string: A boolean expression in literals, with curly braces as matching delimiters and round parentheses used for grouping. There are three allowed positive Boolean operators: AND (&), OR (+) and NEAR (@). There is no unary operator (negation) but there are subtraction operators AND NOT (!&) and AND NOT NEAR (!@). The literals are any string of reasonably normal characters inside of curly braces. This allows whitespace and punctuation to appear in the literals. For exact specifications, see the appendix. Basic instructions appear at the right-hand side of the web interface.

2. From date: Any date from 2012-10-16 to the day before the present day, written in the above format.

3. To date: Any date from 2012-10-16 to the day before the present day, written in the above format. If the range contains days on which data is missing, it will say so. If the date range is flawed (e.g., NOV 1 to 31), then the serach will not run.

4. Number of Excerpts: Out of all the ripped textblocks comprising the total count of instances of the search string, how many should be printed to screen. The box is filled with the default parameter 100, which can be changed.

5. Excerpt size: number of characters forwards and backwards from the search term that define the window which is excerpted. The box is filled with the default parameter 400, which can be changed.

6. Random (checkbox): if left unchecked the displayed excerpts are picked uniformly at random (without duplication) from among all excerpts found. If checked, it outputs the first ones chronologically. [Note: this should be called “non-random”, not “random”.]

1.2 Simple search outputs

1.2.1 Screen output

First comes a summary of the input parameters for the search.

Next, ripped textblocks are displayed, up to the maximum requested.

Next, there is a table whose rows are the days in the search timeframe and whose columns are as follows.

1. `dt`: date
2. `mt_cxt`: number of distinct contexts for the search term. This counts at most one occurrence per program episode, and does not count it if the search striung has already come up in a previous episode in a window extending forward and backward by 20 characters which is verbatim identical to the present occurrence.
3. `mt_prg`: number of distinct programs in which the search term was found. This is, by definition, larger than `mt_cxt`.
4. `tot_mt`: Total number of occurrences of the search string. In cases of complicated Boolean architecture the precise definition is not obvious (see appendix).
5. `sel_prg`: the number of programs on the list of programs over which the search was conducted that aired on that day.
6. `tot_prg`: the total number of programs on the list of captured programs that day. This agrees with `sel_prg` unless a proper subset of programs was used. These two quantities are useful for normalization of the numbers of matches.

The final part of the output is a set of buttons for obtaining download files.

1.2.2 Output files

1. Output CSV is a .csv file containing exactly what is in the table: one header row plus one row for each date plus one row that gives the column totals.

2. All Excerpts CSV is a .csv file with one row for each hit in the Boolean search. The number of rows is thus equal to the column total for `tot_mt` from the previous table. There are three columns: `dt` contains the date in YYYY-MM-DD format; `program` contains the program title (an ASCII unique identifier supplied in the program metadata); `excerpt` is the ASCII text block in which the match was found, extending on either side of the match by the number of characters specified in the "Excerpt size" field.

3. Random Excerpts CSV is the same except it contains only the randomly selected excerpts output to the screen. This randomization is not the same on different runs; the .csv file allows one to capture the specific sample for one run in a format that can be uploaded to R, Excel, and so forth, for text analysis. Also it can be distributed among a group of analysts for multi-coder reliability checking.

2 Multiple search

2.1 Multiple search inputs

All the inputs except the first are the same as in the Simple search.

1. Search strings: Any number of Boolean search terms, as defined in the simple search above, separated by newlines.
2. From date: Any date from 2012-10-16 to the day before the present day, written in the above format.
3. To date: Any date from 2012-10-16 to the day before the present day, written in the above format. If the range contains days on which data is missing, it will say so. If the date range is flawed (e.g., NOV 1 to 31), then the serach will not run.
4. Number of Excerpts: Out of all the ripped textblocks comprising the total count of instances of the search string, how many should be printed to screen. The box is filled with the default parameter 100, which can be changed.
5. Excerpt size: number of characters forwards and backwards from the search term that define the window which is excerpted. The box is filled with the default parameter 400, which can be changed.

2.2 Multiple search outputs

2.2.1 Screen output

Again, a summary of the input parameters for the search comes first.

Next, a table is displayed.

In this search, the table comes before the textblocks. Again, the rows (other than the header) are the days in the search timeframe. The columns are the date columns, followed by one for each Boolean search, labeled by the name of the search and returning the number of distinct contexts found for that search on that day; this is the same number that would show up as `mt_cxt` on that day if one conducted a simple search. A final column is included which is the selected program count for the day; this is the same as the value of `sel_prg` returned by any simple search on that day.

> Note: the rows of the table are output one at a time as they are computed, allowing the user to explore, then stop the search and start a new one.

Next, ripped textblocks are displayed, up to the maximum requested.

The final part of the output is a set of buttons for obtaining download files.

2.2.2 Output files

There are five downloadable output files for the multiple search.

1. Output contexts file: This .csv file is precisely the data in the table just described, with a date column, a column for each search, and a selected programs column; there is one non-header row for each day.

2. Output programs file: Same except that it counts `mt_prg` for each search on each day rather than `mt_cxt`.

3. Output total matches file: Same except that it counts `mt_tot` for each search on each day.

4. Output long file: This .csv is meant to allow comparisons between the three previous measures, `mt_cxt`, `mt_prg` and `mt_tot`. There is a row for each combination of date and search; the five columns are: date, search term, contexts, programs and total matches.

5. Visualization: This is the result of plotting one of the three quantities (contexts, programs or total matches) as N time series (polygonal graphs). Here N is the number of searchs conducted. The default choice is contexts, but it can be changed once the graph is displayed by changing which variable box is checked. There are a number of other choices one can make once the multiple graph is displayed.

    - You can toggle whether the data is displayed as is versus displaying a 7-day moving average; the default is the moving average.
    - You can toggle whether it displays raw numbers or a percentage of the total of all the parallel searches; percentage is the default.
    - For clarity, any one of the N lines may be highlighted by mousing over the corresopnding color segment in the legend to the upper right.
    - If you hold the CTRL key and mouse over a vertex of one of the N line graphs, you see the data associated with that data point (for any of the three measures, not just the one being displayed).
    - You can zoom in on any subrectangle by clicking on "Reset Zoom" at the lower left and then selecting a rectangle within the graph area by clicking on one corner and dragging to the other corner.

3 Matrix search

3.1 Matrix search inputs

The primary purpose of Matrix search is to tabulate how often pairs of terms in a given list occur together, meaning in the same program and nearer than a given character distance. The input parameters are as follows.

1. Search strings: Any number of Boolean search terms, as defined in the simple search above, separated by newlines.
2. Character distance: how near two search hits must be to count as occurring together.
3. From date: Any date from 2012-10-16 to the day before the present day, written in the above format.
4. To date: Any date from 2012-10-16 to the day before the present day, written in the above format. If the range contains days on which data is missing, it will say so. If the date range is flawed (e.g., NOV 1 to 31), then the serach will not run.
5. Number of Excerpts: Out of all the ripped textblocks comprising the total count of instances of the search string, how many should be printed to screen. The box is filled with the default parameter 300, which can be changed.
6. Filter top for visualization: in the visualization, described in the appendix, a limit is usually desired as to how many of the search terms will appear in the display. Putting N in this box instructs the visualizer only to display the top N in terms of number of hits. By default this box is populated by the number 25.

3.2 Matrix search outputs

3.2.1 Screen output

Again, a summary of the input parameters for the search comes first.

Next, an announcement is made as each day's data is processed.

When this is done, the total is printed out for selected programs and total programs, summed over the date range, followed by four tables in awkward tabular printout format:

1. Table 1 has rows indexed by pairs of search terms. The first two columns are the two search terms in question and the other three are the number of successful searches for the pair of terms. If the terms are A and B, this number is the same as what you would get if searched for A @k B where k is the chosen character distance parameter.

2. Table 2 has rows and columns indexed by search terms. The entry corresponding to terms A and B is the number of contexts in the date range for A @k B.

3. Table 3 is the same as Table 2 except that it counts programs rather than unique contexts (thus the value is at least the value in Table 2).

4. Table 4 is the same as Table 2 except that it counts total matches rather than unique contexts (thus the value is at least the value in Table 3).

3.2.2 Output files

The final part of the output is a set of buttons for obtaining seven possible download files and a graphic.

The first six are all downloads of the data in Table 2. The second downloads it directly. The first does so with the header row and column stripped while the third provides the lead column only (in other words, just the search terms).

The next three are the same as the first three except that any search term with a count of zero (on the diagonal, hence the whole row) is removed from the table.

The "Visualization file" has a row for each search term. The first column contains this search term. The second column repeats the first unless the search terms match a dictionary, in which case the dictionary abbreviation is given. The third column gives the "size" of each search, which is the diagonal entry, normalized so that the largest is 10 and the smallest of the ones used is 4. The remaining columns are indexed by the search terms, in the same order. These columns form a distance matrix. The distance between two points is a decreasing function of the number of co-mentions, normalized as a 2×2 correlation matrix. Details are given in the appendix.

Visualization

Finally, the "Visualization" button produces a graphic rendering of the distance graph. There is one node for each search term; if the number of terms is greater than the parameter "Filter top for visualization", then only that many entries are chosen, based on their number of hits (the diagonal entgry in the matrix).

Nodes are sized according to their diagonal entry. Positions are computed by a spring-energy rendering routine. This renders positions in the plane by following a gradient descent algorithm with a quadratic energy function that would be minimized if all edge lengths were exactly equal to the correspoding matrix entry but can typically not reach that because of the restriction to two dimensions.

Mousing over an edge displays the distance matrix entry for that edge. Mousing over a node displays the size entry for that node.

4 Appendix: technical definitions

4.1 Boolean search count

TBA

4.2 Size of node

TBA

4.3 Distance between pair of nodes

TBA