CoVis : Visualizing Character Class Collocation in Ancient Chinese Literature

Paul Bucci

CPSC 547 Final Report

Abstract

When analyzing a very large corpus of texts, it is imperative to use a computational tool to gather statistics about the relationships between words that appear in the corpus. A key statistic is word collocation. If two words are to appear in close proximity, they can be said to be collocated, and therefore conceptually linked. A class of related words can be said to be conceptually linked to another class if members of the class are collocated within a corpus.

This paper describes a visualization tool for aiding in the analysis of collocation data, called CoVis for short. CoVis was developed specifically for supporting a Cultural Evolution of Religion Research Consortium (CERC)[1] project that looks at the relationship between religion and culture in a specific corpus of ancient Chinese literature. This paper will focus on that corpus, however, the techniques described should be generalizable to any textual collocation analysis project.

1.0 Introduction

Analysis of large text corpora has recently become more feasible with the proliferation of data analysis techniques in the humanities. With a sufficiently large corpus, reading a set of texts becomes impossible, therefore, inferences about meaning within the texts must be made on key statistics. This paper focuses on collocation statistics, where two words are said to be collocated if they are in close proximity. The goal of collocation analysis is to determine which words are conceptually linked via collocation. This should be intuitive: if two words are near in a text, they are likely to being used to refer to a common concept.

Collocation can happen within a sentence, or within a certain proximity, called the collocation window. Words that appear in the same sentence are clearly strongly related, and words that are nearest are more strongly collocated. These two collocation dimensions interact, and are not completely separable, therefore they need to be considered both together and separately. Proximity collocation strength is calculated by the frequency two words are collocated within a certain window, and sentence collocation strength is the frequency two words appear together between delimiters.

With some domain knowledge, a researcher will organize related words into classes. For example, “bus”, “train”, and “subway” are members of the *Transit* class; “love”, “good”, and “happy” are members of the *Good emotions* class; “dislike”, “bad” and “angry” are members of the *Bad emotions* class. The sentence “I love taking the train” is an example of a strong collocation of the two classes, therefore, we can infer that this text has conceptually linked *Transit* and *Good emotions*. The sentence “I dislike taking the bus” has conceptually linked *Transit* and *Bad emotions*. With a corpus of tracts on transit preferences, we could infer which emotions are more strongly linked to *Transit* by counting the frequency of collocations within sentences and within a variety of windows.

To perform an analysis, a researcher will identify a focal class and one or more comparison classes. For our above analysis, *Transit* is identified as the focal class, and *Good emotions* and *Bad emotions* are the comparison classes. The analysis will be *Transit → Bad Emotions* vs *Transit → Good Emotions*. For each text in the corpus, collocation frequencies for each *Focal → Comparison* are recorded, as well as the frequency that members of each class occur in the text. From this, other statistics can be derived such as conditional probability for a collocation window, calculated by *(Focal → Comparison frequency within a window)/(Number of focals in the text)*.

The high-level goal of the CERC project is to examine the relationship between high gods, deities, punishment, reward, and morality. Specifically, the project asks the question of whether High Gods exist in the corpus, and, should they exist, whether they enforce morality through reward and punishment. From this, CERC will attempt to establish the historical prominence of dualism in ancient Chinese philosophy. The texts examined are digitized versions of the untranslated originals, which are organized into genres such as History, Math, Etymology, etc.

CoVis was designed to support an exploration of collocation data from an arbitrary grouping of texts and *Focal → Comparison* statistics. Using pre-calculated collocation data presented in the form of a CSV, CoVis calculates a variety of statistics on the fly, and presents the user with a small set of visualizations for collocation strength by text, by set of texts, by word classes, and by time. It further breaks down collocation data by in-sentence counts and counts per window, as well as provides a comparison of the two. This allows a researcher to take an iterative approach to data exploration, starting with a high-level summary, then drilling down into a text-by-text analysis. This process is to help a researcher to identify the interesting questions to ask, and where to explore next.

2.0 Task and data abstraction

2.1 Data

The original data is a corpus of texts and a set of word classes. CoVis was built to visualize the dataset produced by text processing algorithms that were developed outside of this project, and are not included in its direct scope. However, future implementations of this system may need backend processing abilities, and therefore they are discussed here.

The text processing algorithms represent the focal-comparison relationship for each text as a network with focal and comparison classes as nodes, and the distances between them as weighted edges. This closely follows the conceptual model of the relationship between words, and therefore it's tempting to carry the network metaphor through as far as possible. However, the speed of traversing a highly-connected graph is too slow to support fast interaction if weights over large sets of edges need to be collected, therefore this was pre-processed.

The correct data abstraction for the output produced is a multidimensional table with genre, focal → compare, and text name as keys, and date and window/sentence/focal counts as attributes. However, the table that was ultimately produced was a 2D .CSV, where each combination of focal → compare for each text was a separate item, and text name, genre, focal class, compare class, date, and window/sentence/focal counts as attributes. For all counting functions that don't separate collocation windows, the maximum window size of 50 characters left/right is used. All counting is cumulative.

A side note on terminology: since the original texts are in Chinese, most words have a length of one character. However, some words are multi-character of length n, called n-nomials, that refer to the same core concept as a whole. These are treated as occupying a single position in a text. As a result, the terminology of 'word' and 'character' both refer to an n-nomial set of of characters. An English analogy would be "stop motion," which is a noun phrase that carries a different meaning as a set of two words than each word does separately.

2.2 Task

CoVis supports a variety of subtasks, but the main task of CoVis is exploration of the collocation data so that a researcher can discover and identify key focal class, compare class, and text combinations to study further. To do this, CoVis presents summaries of collocation data at a number of levels so the researcher can compare trends of collocation strength per focal-compare across time, across individual texts, and across arbitrary selections of texts.

3.0 Related work

Many text analysis tools use a bag-of-words approach, which often precludes effective meta-analysis using text structure and domain knowledge. Two examples of this approach are IBM's ManyEyes[2] word tree, or the ubiquitous Wordle[3]. Both visualize the relative frequencies of words, however, they do not allow words to be grouped by class.

The approach taken by CERC for this corpus analysis is a variant of topic modelling, with word classes as topics. Typical approaches to topic modelling visualization encode topics by size, either over time or over a set of texts. Tools that visualize inter-topic connections often use a node-link metaphor. Some more advanced tools, such as the PNNL's In-Spire[4] such a variety of visualization methods to better differentiate metrics.

3.1 FacetAtlas/node-link approach

Although the visualization approach FacetAtlas[5] uses is quite different than CoVis, the conceptual approach is similar, in that FacetAtlas is attempting to expose the connections between topics as well as the relative size of topics. FacetAtlas uses a node-link metaphor, where topic clusters are aggregated and inter-topic connections are links. Unfortunately, this approach ignores change in topic systems over time, however, as a static snapshot of topic connectivity, FacetAtlas is very powerful. The evolution of inter-topic connectivity is an important dimension for the analysis CERC wants to do.

3.2 Serendip/individual text view

Serendip[6] shares a similar philosophy to CoVis, as it focuses on the relationship between topics and individual texts, as well as provides a number of overview methods. It does not explore inter-topic connectivity, however, and provides a deep individual text view that CoVis does not. This is mostly due to CoVis not having a direct connection to its original data. Although an individual text view seems like a natural choice, the types of questions CERC was looking at are irrelevant at the individual text level.

3.3 Topic rivers

Both ThemeRiver[7] and its successor RoseRiver[8] are interesting approaches where topic evolution over time is visualized as stacked area charts. Again, this approach does not explore the relationships between topics.

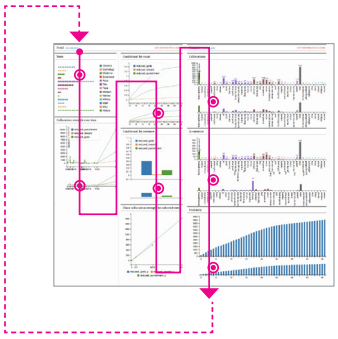
Overview of the CoVis system, which is loaded into a browser.

4.0 Solution

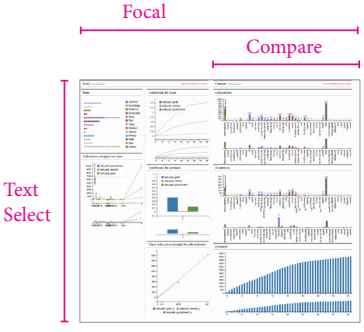
The main design challenge of CoVis was to visualize class connectivity, evolution over time, and class connectivity per text. Rather than attempt a unified approach, CoVis incorporates a variety of widgets that visualize each metric separately.

4.1 Workflow

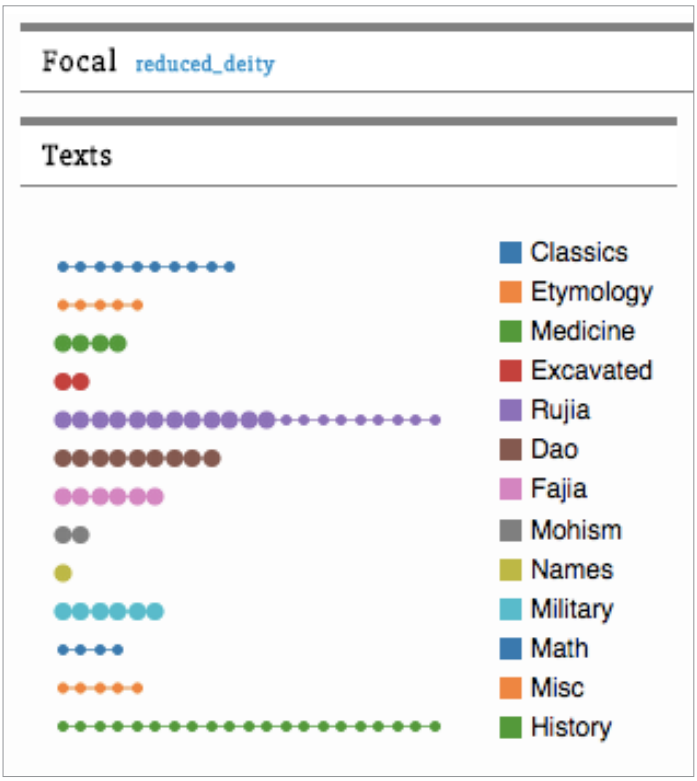
Since this interface is designed mostly for an iterative exploration, the implementation focuses heavily on directing workflow. A researcher will begin by selecting a focal class, then a set of texts, then a comparison class, then iterate. With each selection, the interface updates the visualization widget that the selection influences.



Workflow begins at the top left corner, where a user chooses a focal class and a set of texts to explore.

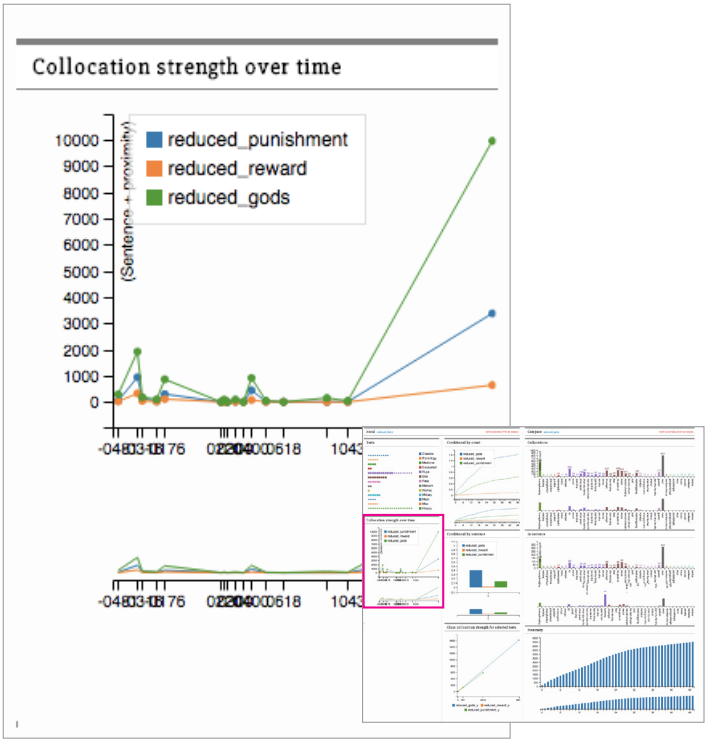


The range of influence of each interaction has a right-to-left precedence.



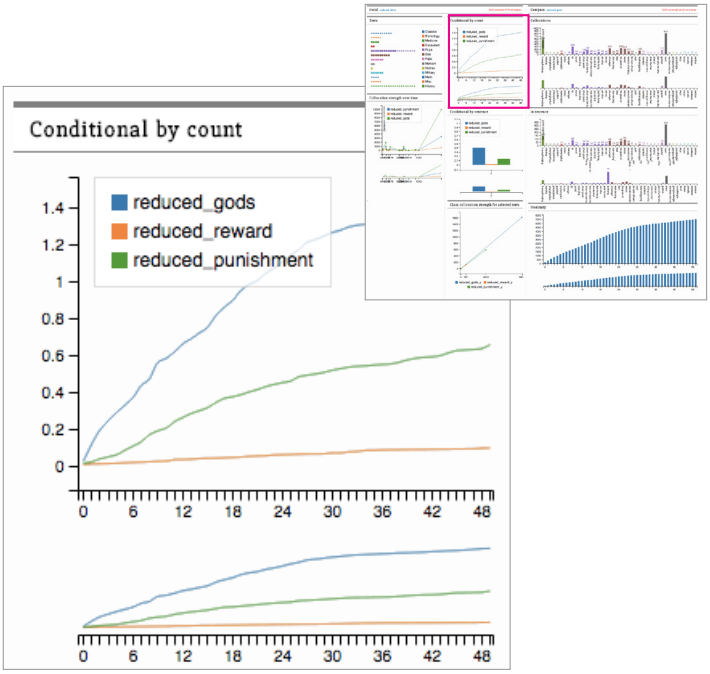
The text selection widget supports a click and drag interaction. Each dot represents a text. A selected text has an expanded dot. Each genre, listed on the right, is also clickable, and will select/deselect an entire genre of texts.

4.2 Interface



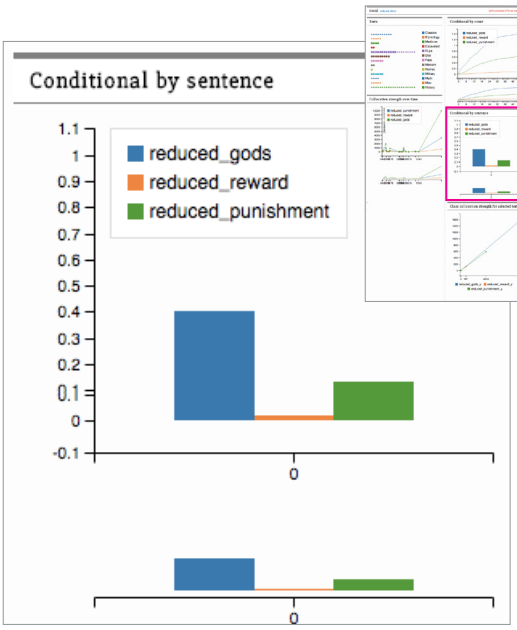
4.2.1 Collocation strength over time

This time series widget encodes collocation strength to each comparison class for a selected focal class as y-position and time as x-position. With this, a researcher can see how collocation strength evolves for all texts over time. The widget also supports zooming.



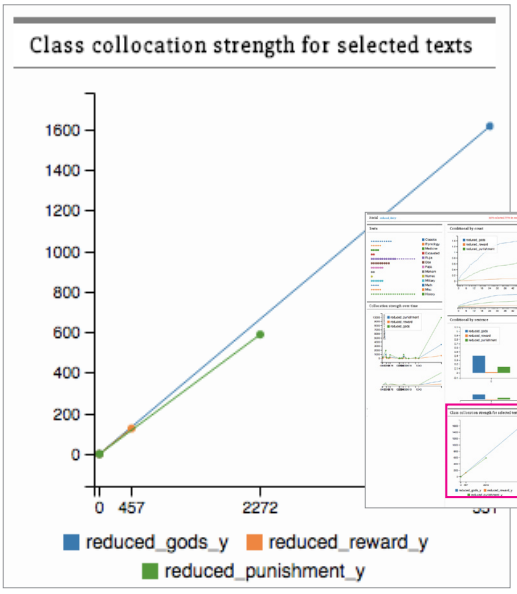
4.2.2 Conditional by count

This widget encodes the conditional probability that characters from the selected focal class to all comparison classes will be collocated within a particular window for all selected texts, with y-position as conditional probability, and x-position as size of window. With this, a researcher can determine very high-level questions such as "In these texts, do high gods punish more than they reward?"



4.2.2 Conditional by sentence

This widget encodes the same as the above, but for in-sentence counts, where y-position encodes in-sentence conditional probability, and colour encodes comparison class. Since there is no evolution for in-sentence counts, a simple bar chart is good for a quick height comparison between comparison classes.



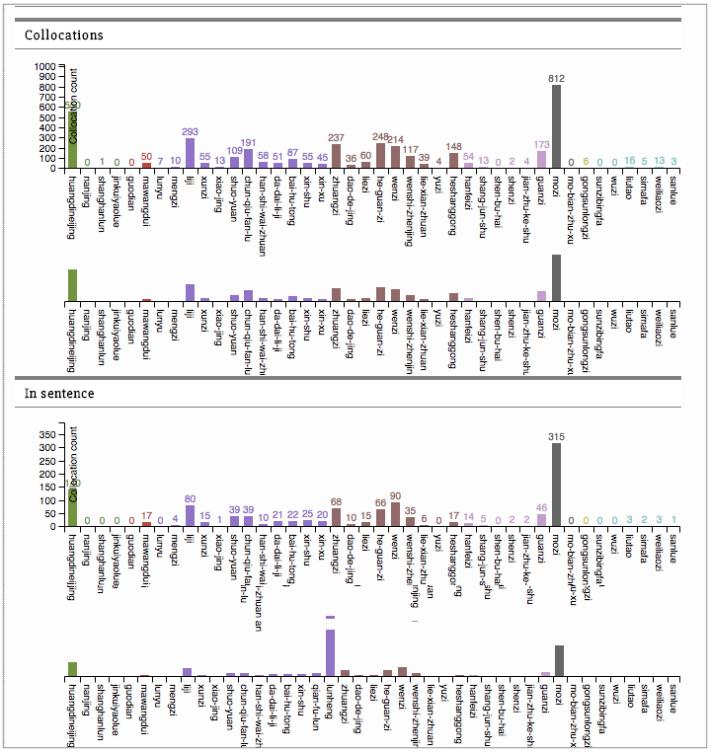
4.2.2 Class collocation strength for selected texts

This widget encodes in-sentence + proximity counts for all selected texts as length by encoding in-sentence counts as y-position, and proximity counts as x-position. By using a line, this allows angle to encode relative in-sentence to proximity ratio as angle. This allows a researcher to quickly compare the strength of collocation, as the maximum class will be a roughly 45-degree angle, and every other class will deviate from there. An angle closer to 90 degrees will show a higher ratio, and the line terminus will be higher than the maximum class line.

Although this encoding is valid for the task of comparing classes, there could be a large variety of alternate encodings. A strong consideration would be a linked circular graph where number of links between classes represent collocation strength. This approach has the unfortunate effect of implicitly encoding strength with area and length, and it could create false impressions of the data. For example, if two classes were next to each other on a graph, it may imply actual closeness in collocation that may not exist. Similarly, links that cross the circle take up more area, and this could create a false impression of importance.

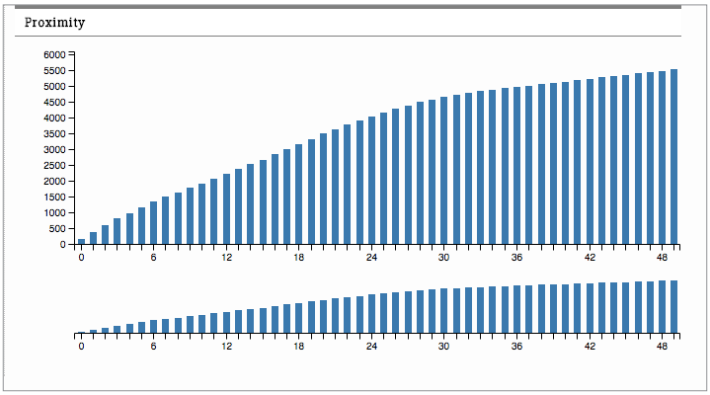
The current approach works, but it misses the metaphor of strength being encoded by properties that imply physical strength, such as closeness, density, or size. It also makes it difficult to compare all classes to each other at once.

The Conditional by count, and Conditional by sentence, and Class collocation strength for selected texts widgets were created to separate the task of comparing classes from the task of comparing texts. This allows the Collocation/in-sentence/proximity widgets to be simple and uncluttered, and also supports a directed workflow. Using these three widgets, a researcher will know which comparison classes to inspect more closely.



4.2.2 Collocations/in-sentence

This widget encodes overall proximity/in-sentence collocation strength by height and genre by colour. This allows a researcher to look across a set of texts at the distribution of collocation strengths and identify key texts to compare.



4.2.2 Proximity

This widget encodes proximity strength for all selected texts for each collocation window. This allows a researcher to identify the key windows to look at in the future. For example, a maximum window of 50 was chosen arbitrarily, this widget will allow a researcher to refine their window.

5.0 Implementation

This project was written in C3js, a chart library for D3js, which is a Javascript library built for manipulating SVG graphics and HTML DOM elements in a browser. It also uses JQuery and a small JQuery dropdown plugin.

My work is completely contained in *functions.js*, *covis.html*, and *style.css*. All functions in functions.js are designed to populate and control the DOM elements declared in *covis.html* and styled by *style.css*. These functions are mostly for reading and interpreting the .CSV database by collecting counts, calculating conditional probability, and loading that data into the widgets.

A quick overview of the system is as such: first, the data from the .CSV is loaded, whereupon the genre-color map, genre list, text list, text-genre list, and class list are created. Each widget is initialized unpopulated. Then, as a user selects a focal class, set of texts, and comparison class, the widgets are updated.

7.0 Results

7.1 Scenarios

7.1.1 Generalized Scenario

A user will begin by selecting a focal class and inspecting the timeline. If the focal class seems interesting, they will then select a set of texts to explore, watching the conditional/class collocation column update. Since selecting texts is quick, they might want to browse through a few permutations. After they've settled on a set of texts to inspect, they use the conditional/class collocation column to inform which comparison class to inspect further. After selecting a comparison class, it may become clear that certain texts or genres are not contributing meaningfully to the data inspection, and the user will go back to deselect uninteresting texts and start the process again.

7.1.2 Specific Scenario

Say the user choses *reduced_deity* as a focal class, and all of the etymology genres. Noticing a cross-over for the *reduced_gods* and *reduced_punishment* lines in the *Conditional by count widget*, they zoom into the cross-over and hover over the point to see a tooltip with the exact numbers. From this, they can conclude that deities are much more likely to be linked to punishment than to reward.

The *Class collocation strength for selected texts* widget tells a slightly a different story. Here, we see that *reduced_gods* characters appear more often in-sentence and in close proximity than *reduced_punishment*. This means that deity and god words appear next to each other often, which makes sense, as deities are less important gods.

After inspecting the two widgets, it's clear that *reduced_punishment* is worth choosing as a comparison class. The *Collocations* and *In-sentence* widgets tell us that there are three texts worth inspecting in more detail. The *Proximity* widget tells us that there is a lot of collocation activity in the 24-character window, then collocations plateau.

Having learned which texts are interesting, the research starts again by deselecting the texts with no collocations and tries another focal class.

7.2 Evaluation

CoVis was developed in consultation with the CERC lab, with Research Assistant Carson Logan as the main point of contact. Logan participated heavily in each stage, starting with negotiating the requirements specification between myself and CERC, informally evaluating design iterations, and performed an informal usability study. Members of the Vancouver Institute for Visual Analytics were also consulted on design idiom choices, with Lab Manager Rama Flarsheim as the main point of contact. Although VIVA members did not evaluate the interface directly, their critical input was much appreciated and incorporated into the CoVis design.

Logan's response was positive. CERC is in the research methods discovery phase, therefore CoVis stands as a proof-of-concept. CERC hopes to scale their text analysis to thousands of documents that span a millennium. CoVis provides a clear example of the power of a custom-built visualization tool for text analysis, as well as the power of supporting a data abstraction that allows for on-the-fly calculations and summaries. At the moment, this type of analysis work is

done line-by-line with tools such as R and Excel, which require a large amount of preparation to create visualizations. According to Logan, CoVis speeds up the exploratory guess-and-check aspect of his work.

8.0 Discussion

8.1 Strengths

CoVis supports the project goals stated in that it addresses mid-stage research questions. Researchers at CERC are in an exploratory phase, where they are familiar with their data, but are looking for new and interesting questions to ask next. Similarly, because they are trying to solidify their research methods, this tool gives them a clear example of what a mid-stage exploration can actually look like.

The workflow developed supports a natural interaction with the data, as it starts at a high level and allows a user to drill down, then iterate. The main interaction space is the text selection widget. Although individual texts are not a primary focus of CERC research, arbitrary text sets are, and this widget allows for quick click-and-drag selection of texts.

The interface is designed to reduced cognitive load while working. Each widget column is separately scrollable, allowing a user to flexibly display the most pertinent widget for the question they are exploring, however, the focal and compare class headers are fixed. This was reconfigured from a previous approach that kept widgets in separate tabs.

8.2 Weaknesses

There are comparison tasks that could be combined in this interface. For example, the collocations and in-sentence widgets are highly repetitive, and might need to be combined or re-oriented such that comparison doesn't require such a leap. The current approach was chosen as it allows for text names to be displayed with ease, however, this might not be a very important feature.

The brushing and zooming is not linked between widgets. This would be a desirable feature, as setting a tight collocation window in the proximity widget should perpetuate throughout the counting widgets.

Some colour encodings use the same colour to encode different data, which is a distraction. The colour palate is hard-coded as a list, which means that there is an upper limit of the number of different colours that can be used. Some sort of automatic palate function would be desirable.

As mentioned above, the current encodings work, but may not be using physical metaphors strongly enough. Further exploration is required for this domain.

8.3 Limitations

At the moment, the data processing happens in-browser, and does not support persistent data. Although the database is pre-computed, implementing a set of dictionaries that are populated as the user interacts would speed up load times. With a small database, this is not an issue, but scaling up to thousands of texts with thousands of permutations would be very difficult.

Another limitation is the .CSV format itself. The data is naturally a multidimensional table, and should be represented as such. This would require a more intensive initialization stage.

The browser as a front end is both powerful and limiting. A strong front to back end interface could be built for the CERC lab such that word classes could be updated and changed, or new texts could be added.

8.4 Future work

Currently, the conditional probability functions are not robust. A request feature from a CERC supervisor was to add an ability for researchers to add their own functions and compare them over time. This would be very interesting, as it would allow researchers to weight their calculations on the fly. It would also imply that a chart-creation function could be implemented.

Although the system was built for a very particular dataset, it would take very little to visualize any dataset that is formatted with the same columns as the current .CSV. This feature hasn't been tested, but it implies that CoVis could be hosted on the CERC website where users could upload their own datasets.

Lessons learned

Coming from a news design background, the first impulse is to create an interface with a lot of negative space, low complexity, and visualizations that take no effort to understand. Although these are still admirable goals, reducing cognitive load after an initial learning period is a much more important goal. Since power and complexity tend to be correlated, there is a tradeoff between power and simplicity. Making this tool powerful enough and easy enough to use introduced clutter and complexity. This tradeoff was especially clear with the limitation of the number of pixels on the screen, as the design had to be densified to accommodate clarity.

9.0 Conclusions

Although this tool was built for a specific dataset, it provides an insight into possibilities for textual analysis in the humanities in terms of workflow and research methods. As a proof-of concept, this project shows that tools such as D3js and C3js enable very flexible and rapid visualization development for

already-computed datasets. This approach abstracts textual analysis from the original texts, allowing high-level analysis.

Note: the working prototype for CoVis can be accessed at http://paulbucci.ca/covis/covis.html.

10.0 References

1. http://www.hecc.ubc.ca/cerc/
2. http://www-969.ibm.com/software/analytics/manyeyes/
3. http://www.wordle.net/
4. http://in-spire.pnnl.gov/
5. Cao et al. Facetatlas: Multifaceted visualization for rich text corpora. Visualization and Computer Graphics, IEEE Transactions on 16, no. 6 (2010): 1172-1181.
6. Alexander et al. Serendip: Topic Model-Driven Visual Exploration of Text Corpora. VAST 2014 Conference Paper.
7. Havre et al. Themeriver: Visualizing thematic changes in large document collections. Visualization and Computer Graphics, IEEE Transactions on 8.1 (2002): 9-20.
8. Weiwei Cui, Shixia Liu, Zhuofeng Wu, Hao Wei. How Hierarchical Topics Evolve in Large Text Corpora. TVCG 20(12):2281-2290 (Proc. InfoVis 2014) 2014.