

Augmenting Citation Chain Aggregation with Article Maps

Timothy Cribbin

Centre for Intelligent Data Analysis, Brunel University, Uxbridge, United Kingdom

{timothy.cribbin@brunel.ac.uk}

Abstract. This paper presents Voyster, an experimental system that combines citation chain aggregation (CCA) and spatial-semantic maps to support citation search. CCA uses a three-list view to represent the citation network surrounding a ‘pearl’ of known relevant articles, whereby cited and citing articles are ranked according to number of pearl relations. As the pearl grows, this overlap score provides an effective proxy for relevance. However, when the pearl is small or multi-faceted overlap ranking provides poor discrimination. To address this problem we augment the lists with a visual map, wherein articles are organized according to their content similarity. We demonstrate how the article map can help the user to make relevant choices during the early stages of the search process and also provide useful insights into the thematic structure of the local citation network.

Keywords. Citation index, chaining, search user interface, document similarity, visualization

1 Introduction

Citation cycling is a powerful strategy that involves navigating backwards and forwards through citation chains in order to build up a picture of the intellectual base around a topic [2]. Generally, this is still achieved using hypertext interfaces provided by digital libraries such as Web of Knowledge and Scopus. However, this page based approach to navigation is problematic for two reasons. First, the size of the search space (cited plus citing articles) around even a single article might be very large. Secondly, as the user navigates step-wise through the network it can be difficult to maintain a sense of context and to form a coherent mental model of the intellectual space unfolding around their interest.

Scientometric research has shown how visualizing citation networks can provide useful insight into the intellectual structure of a field or discipline of research. However, attempts to support citation search using similar maps have achieved mixed results [1][7]. A pervasive issue associated with most approaches to visualizing citation graphs is one of controlling visual clutter. The sub-graph surrounding just a few related articles can be large and complex. Representing such a graph neatly and useably in iconic format is problematic enough, but effective navigational support also requires details in context. In other words, supporting citation chain navigation direct-

ly using maps is not just a “*standard graph visualization problem*” [1], as the user needs simultaneous access to both article content and inter-article relationships. Typically, this limits the view to a single focus along with a graphical representation of its first-order or, at most, second-order relationships, usually in a single direction (cited or citing). Highly connected articles (i.e. the most interesting ones) present a particular problem, with most systems suffering from excess clutter when citations exceed just a few dozen.

In 2011, we proposed a novel approach to citation cycling called Citation Chain Aggregation (CCA: [4]). Rather than attempt to visualize citation chains graphically, article records are instead displayed textually within a three-list view (see Fig. 1). Known articles are displayed in the central *pearl* list, whilst their cited and citing articles are displayed in the left and right views respectively. As the user adds articles to the pearl list, the peripheral lists are updated by adding new articles and, most importantly, by registering shared citations and references by means of a displayed *overlap score*. This overlap score provides a powerful cue to relevance, based on the logical assumption that for a given set of known relevant articles, if *article A* cites or is cited by more of those articles than *article B*, then it is more likely to be relevant to the user’s interest.

However, there are some circumstances in which overlap scores do not provide sufficient cues to relevance. This is especially true for sessions that begin with just one known article, particularly if this is highly cited. It is also a potential problem for more complex searches where the pearl relates to multiple aspects of a broader topic.

To resolve this, we return to the idea of using a visual map as a means of augmenting the functionality of the CCA lists. Rather than visualizing the citation network itself, we followed a similar approach to tools like CAVis [8] and RefViz [5] where article nodes are arranged according to content similarity. We demonstrate how this new functionality can address the issues described above, before discussing future avenues for development and application.

2 Citation Chain Aggregation

A key inspiration behind the CCA model was Larsen’s Boomerang algorithm [6]. Boomerang automates the citation cycling process, eliminating the need for the user to identify good seed articles and navigate the resultant citation graph. The process begins with the user querying multiple indexes (e.g. title, keyword, abstract) with the same query (Step 1). The algorithm then constructs the sets of citations made by each of these retrieval sets (Step 2) and then forward chains (Step 3) from any citations that appear in more than one Step 2 set. They found that the precision of articles found at Step 3 was improved over the original Step 1 sets when articles were ranked according to the degree of set overlap at Stage 2.

CCA [4] is based on the same notion of cognitive overlap or the union of citation sets relating to multiple relevance exemplars. However, it differs from Boomerang by being a more interactive and open-ended process. It draws upon the well-known strategy of citation pearl growing, whereby the user might begin with just one or two rele-

vant exemplars, exploiting the common attributes of these articles to refine their query and thus retrieve further relevant examples. In the case of CCA, the attributes exploited are the common citations or citing articles.

The fundamental concept in CCA is the *citation chain*. This is defined as the sub-graph describing an article and all of its immediate ancestors (cited articles) and descendants (citing articles). In the conventional hypertext model, the user can only view (part-of) one citation chain at a time. CCA, on the other hand, aggregates multiple citation chains into a single three-list view (Fig. 1).

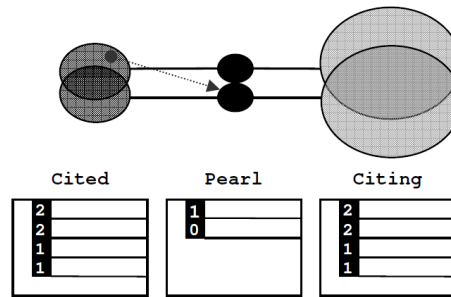


Fig. 1. Citation chain aggregation example, using two pearl articles

A citation cycling episode begins with the researcher adding one or more relevant articles to the pearl list. Each time an article is added, the system sends a query to the citation index web service (currently the Microsoft Academic Search API¹) to retrieve all cited and citing articles. If an article does not already exist as a record in the local *CCA index* then a new item is added, otherwise the existing record is adjusted to reflect another instance of overlap between citation chains. Citation chains are recorded by means of *isCited* and *isCiting* lists attached to each article record. These lists store local key sets that point back to associated pearl articles. The overlap score of any article is simply the number of items in the relevant (*isCited* or *isCiting*) list.

Once all additions or adjustments have been made to the index then the display lists are updated. Each row in a list represents a single article and each article can only appear once in any list. By default, the lists are sorted by descending overlap score, thus giving precedence to articles that cite or are cited by the most pearl articles, although a list can be sorted by any other attribute (e.g. citation count) if required. In Fig. 1, for example, articles with an overlap score of two appear at the top of the peripheral lists as they relate to both of the pearl articles. As items are added to the pearl, overlap becomes a more effective discriminator and proxy for relevance [4]. Note that pearl list items themselves also display an overlap (cited by) score which provides a salience ranking within the known articles set (e.g. in Fig. 1. one article is cited by the other but not vice-versa).

¹ <http://research.microsoft.com/en-us/projects/academic/>

In both the existing Oyster tool² and the Voyster prototype, citation relationships are presented dynamically in response to user interaction. Selecting a pearl item highlights (in bold) all related items in the peripheral lists. Likewise, selecting a cited/citing item highlights related pearl articles in the central list. Lists can be sorted by any attribute and the user can drill down into article details by clicking on a list item. A left-button click displays article metadata in the *Article Summary* pane (see Fig. 2), whilst a right click brings up a menu with options to add/remove articles and also to navigate to the full-text.

There are two main use case scenarios. A cold start scenario is where the user is investigating a new topic. Here, the process begins with the user performing a keyword search (using a search UI built into the tool) and then growing a pearl from just one or two items selected from the results. A hot start, in contrast, is a scenario where the user already possesses a relatively mature bibliography on their interest and wishes to identify salient gaps (e.g. new articles or items missed during earlier searches). A specific example might be checking the completeness of a reference list prior to submission (or during review).

Early trials have indicated that users found Oyster to be a useful tool. However, many users wanted some way of narrowing their search space, particularly during the early stages of a cold start task when overlap discrimination is poor. One solution is to provide filters (e.g. keywords, venues, date ranges). However, users also expressed an interest in the idea of using visualization to help them understand the thematic relationships between articles. As we already had experience in the area of spatial-semantic document visualization, it seemed natural to explore how such a technique might benefit the CCA process.

3 Voyster System Description

Voyster is a direct evolution of the Oyster tool that has been available as freeware since 2012. Both tools now rely on the Microsoft Academic Search database, rather than Web of Science (WOS) [4], which means they can now be used without subscription. Although, Microsoft ceased updating the index in sometime in 2011, it is still a substantial database comprising some 38 million articles spanning around 15 different disciplines and web service performance remains sufficiently fast and reliable for evaluation purposes.

The significant change over the Oyster system is the addition of a dynamic, interactive article map (see Fig. 2). This spatial-semantic visualization is a projection of content similarities expressed between all articles currently displayed within the lists. The map is dynamically linked to the list views, such that selection of an article in one view is reflected immediately in the other.

² <http://people.brunel.ac.uk/~cssrtfc/oyster.html>

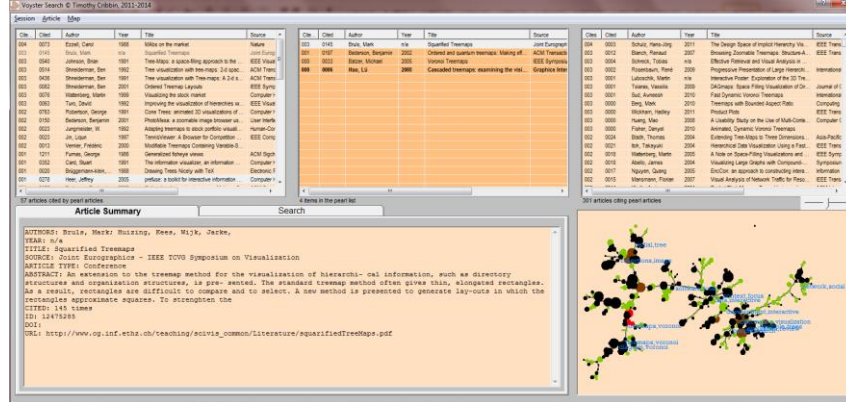


Fig. 2. Voyster User Interface. The original CCA UI has been extended by adding the spatial-semantic map in the bottom right corner.

3.1 Creating the Article Map

Voyster employs a bag-of-words approach to text analysis. Each time the index is updated, a word-document matrix is created based on title and abstract contents, from which a similarity matrix is then derived by computing the cosine (normalized dot-product) between all document vector pairs. The similarity model is prepared for layout by pruning the complete graph down to its minimum-spanning tree (MST). This step both simplifies the layout process whilst producing a more trustworthy and aesthetically pleasing visualization [3].

The whole map creation process is very efficient. A sparse vector model is used to store all matrices. Spatial layout is achieved using the fast *sfdp* scaling algorithm provided by the GraphViz package³. As a result, a map of around 300 nodes (see Section 4) can be created and displayed in less than 5 seconds and using less than 25 MB of memory.

3.2 Presentation and Interaction

Once generated, the map is presented using a custom control, which handles scaling, rendering and interaction tasks. Fig. 2 shows how the map has been incorporated into the bottom right corner of the UI. Articles are represented as circular nodes organized using a spatial-semantic or distance-similarity scheme. This scheme means that similar articles tend to form distinct clusters within the map. The grey links further support navigation by emphasizing the most salient inter-article relationships.

Both size and color are used to encode attributes. By default, all nodes are of equal size. However, it is often useful to encode measures of article salience (i.e. citation count or overlap score) into size, as we demonstrate in Section 4. Citation count is encoded as a log-value to prevent high impact from obscuring the view.

³ <http://graphviz.org/>

The default node color is black, but nodes can change color in response to article status and user interaction. Pearl nodes are always shown in red. When a pearl article is first selected (either in the list or map) related nodes change to brown (cited) or green (citing) depending on their status. A mouse-over event will change the node outline to white, whilst a left-click selection will change the outline to yellow.

Animation is also used to help the user to recognize change and relationships within the map. Selecting any article (list or map) will cause its node to pulse i.e. expand and then gradually shrink back to original size. If the article is not a pearl, then this initial pulse is followed by a series of pulses of related pearl nodes, so that the user can easily visualize the overlap relationship within the map context. The user can also smoothly zoom and pan within the map, which can be particularly useful as the index increases in size.

Labels are also rendered to provide context to the map. Nodes are nominated on the basis of their degree (linkage) within the MST graph, a threshold that the user can control to increase or decrease the density of presented labels. Label terms are defined by selecting the two top ranked terms occurring in the document vector.

Note that the coordination between lists and map works in both directions, such that selecting any article node will highlight the article item in one of the lists. Hence there is always only a single article selection/focus, with corresponding metadata displayed in the article summary pane. A mouse-over event in the map also triggers a tool-tip displaying the article title, which further aids rapid browsing.

4 System Walkthrough

In this section we demonstrate how the map can be used to support a typical cold start search task. Bob, our user, is interested in applying *Treemaps*, the space-filling approach to tree visualization invented by Shneiderman’s team in the early 1990s. Although the technique proved popular in various applied domains, One problem with the original algorithm was that the slice and dice approach led to nodes becoming too elongated and thus difficult to compare and select. Bob has been told about a refinement called “Squarified Treemaps” (Bruls, 2000), which is able to preserve a more balanced aspect ratio across nodes, but wants to know what else has been done to improve the legibility of Treemap visualizations.

Bob does a search for Bruls’ paper and adds it to a new pearl list. This first iteration results in a CCA index of 150 articles – 1 pearl, 8 cited and 141 citing. All overlap scores default to one at this stage, so are of little help. He therefore turns to the article map, selecting the option to encode citation count into node size and then selecting the pearl node to produce the map shown in Fig. 3.

The first notable feature is that the cited articles (brown nodes) are relatively dispersed. The largest cluster, just below the pearl (red) node, relates to earlier/original Treemap research and is central to the paper, whilst the other citations relate to various tree visualization techniques that are mentioned only in passing by the authors, to provide historical context.

The vast majority of articles, however, are citing (green nodes). Interestingly, the most salient cluster lies adjacent to the bottom right of the pearl article. The nearest neighbor is Balzar's 2005 paper on "Voronoi Treemaps", cited 33 times. Foraging locally around this cluster subsequently reveals a rich array of other techniques for controlling aspect ratio and other legibility issues, including Cascaded, Ordered and Quantum treemaps. Bob decides to add the most cited (197 times) article: Bederson's 2002 paper on Ordered and Quantum Treemaps.

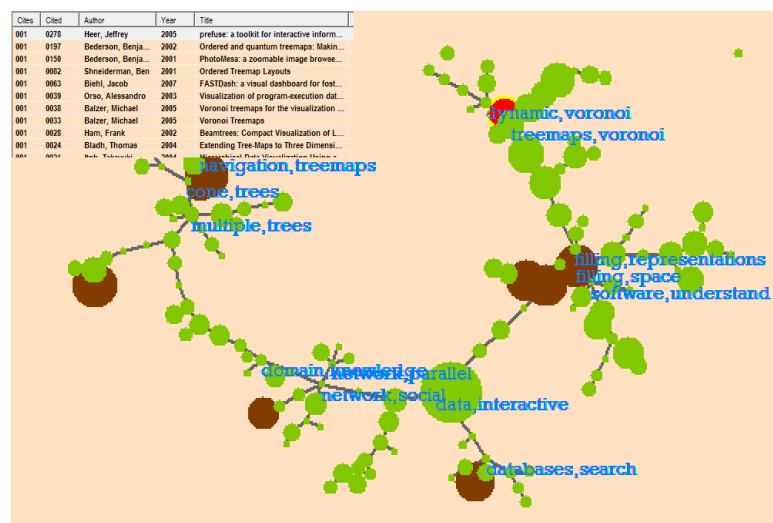


Fig. 3. Map of the Citation Chain around a single article (Bruls, 2001). Node size encodes citation count. Inset box shows the top rows of the citing list.

This action adds 14 articles to the cited list and 145 articles to the citing list. There are now 304 unique articles in the CCA index. The richest pickings still appear to reside within the citing subspace. However, some 46 out of the 286 citing articles cite both articles so overlap score is still a weak discriminator. Bob turns once more to the map, which has now been updated to incorporate the newly retrieved articles, for further guidance. He notices that the two pearl articles are joined by a path containing just two nodes (Fig. 4, left). One of these is Balzar's Voronoi paper, whilst the other is Hao's 2008 paper on Cascaded Treemaps. Impressed by the stability of the local map structure, Bob decides to add the Voronoi article to the pearl.

At this point, the overlap scores are starting to discriminate more clearly between the articles in the citing list, with 8 three-overlap articles appearing at the top of the list, followed by some 50 two-overlap articles. Bob encodes citing overlap score as node size (see Fig. 4 right) to help him choose from these promising cases. The two largest (3-overlap) nodes in the vicinity are further papers about the Voronoi method.

user to stay focused and make rapid relevance judgments during the early stages of the citation cycling process when overlap scores provide poor discrimination. We have also indicated the potential utility of article clustering for exploring broader aspects relating to the user's interest.

Our demonstration shows how local structures tend to remain reasonably stable from one iteration to the next. However, this stability cannot be guaranteed. Currently the user can only one add article per iteration, forcing a choice that could result in difficulty relocating other interesting articles later. One solution is to allow the user to multi-select and add articles. Another (possibly complementary solution) might be to provide the facility to bookmark articles without committing them to the pearl.

Although the map generation is reasonably fast for small corpora, its usable limit is currently reached at around 1000 or so articles. Implementing faster (e.g. multi-core or GPGPU based) algorithms can address this problem, along with the provision of filtering functionality (by keyword or map cluster) to prune the index of less relevant articles would enable users to better manage the size of the CCA index.

Finally, in this paper we have focused on citation searching. Future work is planned to explore how Voyster might be used to support more analytic tasks including research planning and evaluation. For instance, visualizing the citation space around an author or research group's oeuvre (i.e. a form of the 'hot' start scenario) may be a useful means of identifying unexpected areas of impact or for characterizing qualitative and quantitative changes in impact over time.

References

1. Bergström, P., & E. James Whitehead, J. (2006). *CircleView: Scalable Visualization and Navigation of Citation Networks*. In the Proceedings of the 2006 Symposium on Interactive Visual Information Collections and Activity (IVICA), College Station, Texas.
2. Cawkell, A. E. (1998). Checking research progress on 'image retrieval by shape-matching' using the Web of Science. *Aslib Proceedings*, 50(2), 27-31.
3. Cribbin, T. (2010). Visualising the structure of document search results: a comparison of graph theoretic approaches. *Information Visualization*, 9(2), 83-97.
4. Cribbin, T. (2011). *Citation Chain Aggregation: an interaction model to support citation cycling*. In the Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11), Glasgow, UK.
5. Glassman, N. (2005). RefViz 1.0.1. *Journal of the Medical Library Association*, 93(2), 293-294.
6. Larsen, B. (2002). Exploiting citation overlaps for Information Retrieval: Generating a boomerang effect from the network of scientific papers. *Scientometrics*, 54(2), 155-178.
7. Mackinlay, J. D., Rao, R., & Card, S. K. (1995). *An organic user interface for searching citation links*. In the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95).
8. Nguyen, Q. V., Huang, M. L., & Simo, S. (2007). *Visualization of relational structure among scientific articles*. In the Proceedings of the 9th International Conference on Visual Information Systems (VISUAL '07), Shanghai, China.