Typograph: Multiscale Spatial Exploration of Text Documents

Alex Endert, Russ Burtner, Nick Cramer, Ralph Perko, Shawn Hampton, Kristin Cook

Pacific Northwest National Laboratory
Richland, WA USA

Abstract— **Visualizing large document collections using a spatial layout of terms can enable quick overviews of information. These visual metaphors (e.g., word clouds, tag clouds, etc.) traditionally show a series of terms organized by space-filling algorithms. However, often lacking in these views is the ability to interactively explore the information to gain more detail, and the location and rendering of the terms are often not based on mathematical models that maintain relative distances from other information based on similarity metrics. In this paper, we present Typograph, a multi-scale spatial exploration visualization for large document collections. Based on the term-based visualization methods, Typograh enables multiple levels of detail (terms, phrases, snippets, and full documents) within the single spatialization. Further, the information is placed based on their relative similarity to other information to create the "near = similar" geographic metaphor. This paper discusses the design principles and functionality of Typograph and presents a use case analyzing Wikipedia to demonstrate usage.**

Keywords —Visual analytics, sensemaking, text analytics, spatialization

I. I. INTRODUCTION

Central to many visual analytic technologies is the ability to foster sensemaking – the cognitive process of gaining insight from the analysis of information [1]. Sensemaking consists of a combination of cognitive stages that include foraging information, determining important subsets, internalizing that information, then externalizing the insights again [2]. In particular, this process highlights the fluidity of these cognitive stages, where users connect information at different levels of scale to produce a coherent understanding. For example, gaining insight into a collection of text documents requires understanding the relationships between keywords and documents to ultimately create a coherent higher-level understanding of themes or topic areas within the dataset.

Spatializations are effective visual metaphors for sensemaking, both through manually generated layouts as well as computationally generated views. For example, Andrews et al. have shown that through manually creating a spatial layout of text documents, users are able to externalize their insights into the layout they create, affording them the functionality necessary to complete intelligence analysis tasks [3]. Similarly, computationally generated spatializations of text documents, such as the IN-SPIRE Galaxy View [4], show users relationships within a dataset spatially. Fundamental to both manual and computation approaches is the geography metaphor (i.e., "near is similar") used to represent the similarity within the information [5].

The sensemaking process for users leveraging such spatializations involves transitioning between document-centric to concept, or term-centric conceptualizations of the information [3, 6]. Thus, while a general trend moving from foraging details to synthesizing higher-level themes or topics exists, sensemaking requires access to multiple levels of detail of the information. For example, Kang et al. present a view on sensemaking that focuses on this continuous fluctuation between overview and detail information scales for analysis [7].

In this paper, we present Typograph (shown in Figure 1, a multi-scale spatialization for text analytics. Typograph approaches the challenge of transitioning between multiple levels of scale during spatial text analysis. It is based on a single, computationally-generated spatialization of text documents, represented as terms, phrases, snippets, and full documents. As such, the design principles of Typograph include:

- Generating a multi-scale spatialization that embeds multiple levels of detail (terms, phrases, snippets, and documents) in a single spatial layout
- Preserving similarity through relative distances between information (i.e., similarity conveyed through relative proximity).

We discuss our approach towards these design principles through a system overview, detailing the computational model to handle large data scales. The functionality is highlighted by a use case, and a discussion of how such a multi-scale approach may be extended in future work.

II. RELATED WORK

A. Text visualization

Research in visual text analytics has produced a wide range of methods to extract keywords, entities, and sentiment from unstructured text. These computationally-generated characteristics of text are the underlying structure upon which many visualizations are constructed.

Advancements in automatic keyword and entity extraction techniques provide a valuable foundation for spatialization and visualization of concepts. Techniques include analysis of n-grams [8], phrases [9], parts-of-speech [10], and hybrid models using both unsupervised and supervised machine learning approaches on individual documents and corpus wide computation. For example, Rose et al. show how entity extraction using RAKE (Rapid Automated Keyword Extraction) can produce keywords and phrases that are relevant and meaningful to users from a collection of documents without any prior training or supervision [9]. Similarly, other open-source entity extraction techniques are available that produce keyword vectors as quantitative representations of text useful for visualization and analysis purposes (e.g., Lingpipe [11], GATE [12], etc.).

Visual text exploration can be performed spatially. That is, the structure extracted can be used to construct spatializations that are similar to the cartographic approach where nearby information is similar [5]. Such an approach is based on the ability for users to understand similarity and relationships between the information based on their relative distance from each other. Computationally, visualizations such as IN-SPIRE [4] produce views that place documents into a spatialization based on performing dimension reduction on the high-dimensional term vector representations of the documents.

978-1-4799-1293-3/13/$31.00 ©2013 IEEE

Figure 1. Typograph is a spatial text exploration visualization that shows multiple levels of detail within a single spatialization to enable multiscale visual data exploration.

These spatializations rely heavily on the statistically generated signals that derive the relationships and similarities formed. Previous work has also been performed on allowing users to augment these computationally-generated spatializations with valuable domain expertise of users. For example, work has been done to show how users can steer popular dimension reduction methods used for text analysis (as well as other domains) [13, 14]. These semantic interaction techniques enable users to inject their domain expertise into the computational pipeline used to generate spatializations of high-dimensional data through capturing and inferring common analytic interactions (e.g., highlighting, searching, grouping of documents, etc.) [15].

Several other document-centric spatializations exist (e.g., [16-18], etc.) that emphasize the relationships between documents based on the keywords or phrases. Typograph differs fundamentally from these approaches, in that the spatialization presents users with terms, where the documents define the relationships and similarities. The relative distance between terms is defined by the similarities calculated using the documents. Spatial layouts of documents can also be used to provide context around search results. For example, Nocaj and Brandes describe an approach for displaying search results within a spatialization of the document corpus [35]. As a result, users can be given query results in the context of the dataset.

Typograph creates a hybrid between document-centric and term-centric spatializations (described in section III).

B. *Term-centric Spatializations*

Also frequently referred to as "tag clouds", "word clouds", and "Wordles", these spatializations visualize a set of relevant terms in a spatial layout. Fundamentally, their construction focuses on the reduction of whitespace within a bounded area, leveraging various typesetting and packing techniques [19-21]. Font sizes and color ramps are popular visual encodings for conveying relative importance or occurrence counts of terms within the dataset. These views provide a quick, quantitative overview of terms within a dataset.

There has also been work on extending the design of these spatializations to include semantically meaningful placement of terms. For example, Cui et al. presented a context preserving word cloud that maintains the design principles of a word cloud while leveraging similarity metrics between terms to place similar terms near each other [22]. Their approach uses a force-directed algorithm to place terms near each other based on calculated similarity between terms represented as the edges (or springs) pulling terms together. Similarly, ProjCloud uses multidimensional scaling to determine the placement of terms [23].

C. *Sensemaking through Spatializations*

Sensemaking is a cognitive process that highlights a user's ability to forage, synthesize, and externalize information [2, 24, 25]. The process highlights the importance of transposing a user's personal experiences and domain expertise onto a dataset or problem, gathering and evaluating evidence for multiple hypotheses, and ultimately externalizing the insights into a concise analytic product (e.g., a report, presentation, etc.). Pirolli and Card present a depiction of this process as the "sensemaking loop" [2]. The process starts with foraging for evidence and details, and ultimately including synthesis steps that involve determining relationships and higher-level connections within the dataset. Kang et al. show a similar process, but emphasize that through observing such a process in practice, the progression towards insight involves fluctuating between high and low-level concepts (or overview and detail) [7]. It is through such processes that users are able to understand complex relationships within data.

Previous work has shown that a spatialization can be an effective visual metaphor to support sensemaking. Andrews et al. showed that users can leverage the spatial layout as an external memory aid for intelligence analysis tasks [3]. In their study, users created visual clusters of information to externalize their insights. The analysis of such manually created spatial layouts show that the clusters formed are often based on multiple levels of detail [6]. That is, while some clusters may be created based on single keywords or phrases, others may refer to higher-level concepts, or very specific events or hypotheses. As such, enabling users to navigate (and create) spatializations that incorporate multiple scales is important to foster sensemaking.

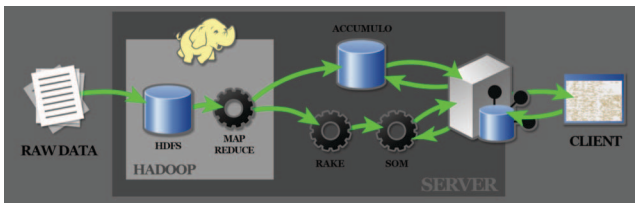

Figure 2. Typograph employs a series of server-side data processing techniques to process high volumes of data and enable interactivity and exploration.

Table 1. The different information types in Typograph extracted and stored on the server using different techniques

Level of Detail	Description	Examples from Wikipedia Dataset	Extraction Technique	Storage Method
Keyword/Term	A word extracted from a text corpus	album, government, baseball	RAKE	Lucene
Phrase	Two or more words in sequence	United States of America	RAKE	Lucene
Snippet	An extracted portion of an article that contains highly relevant information based on a keyword or phrase	Snippet for "World Population" article relative to the keyword "population": Population and Development, the United Nations Population Division, and the United Nations Population Fund. Birth control countries […]	JIL	Extracted at runtime from the articles
Article	Plain text, full articles	A text-only Wikipedia article	Regular expressions parallelized through Hadoop	Accumulo

III. TYPOGRAPH: SYSTEM DESIGN AND DESCRIPTION

The design of Typograph is intended to enhance a user's ability to make sense of large collections of unstructured text. This section describes the design principles fundamental to the system, describes the implementation and primary functionality, and presents a use case to illustrate the usage.

Typograph aims to enable spatial exploration of text documents. As such, there are two primary design principles that drive the visualization and functionality in the system.

***Maintain Relative Spatial Similarity**:* The single, spatial view of Typograph consists of information that is spatialized. That is, all information abides by the "near = similar" geographic metaphor [5]. As a result, information that is similar relative to other information is placed close together, creating clusters of similar information.

In addition to the proximity of information related to other information, the existence of whitespace within the spatialization is equally important. As previous work has shown, the whitespace users create and observe in a spatialization can be as meaningful as the clusters of information [3, 6]. In addition to delineating the clusters themselves, whitespace can indicate areas where coverage of a particular topic is limited, or even missing. For example, whitespace between two information clusters may indicate that the dataset is lacking the information needed to tie the two together. In these situations, the spatial proximity creates the validity in the spatial layout.

***Incorporate Multiple Levels of Detail within Single View**:* Text can be computed and represented at multiple levels of detail (Table 1). Typograph represents the information using keywords, phrases, snippets, and full documents (ordered by level of detail, from highest to lowest). These levels of detail can be accessed by zooming into a specific area of interest from the overview. Through zooming, more detailed representations are progressively shown.

The components that make up Typograph include processing the raw text data for keywords, generating the spatial layout, and the visual interface. These components are discussed in the subsections below.

A. Server-Side Data Processing

For our server, we utilize a 7-node cluster, one of which is the head node. Each node has (4) 4-core Intel Xenon processors with 20Gb memory (except the head node, which has 48Gb of memory).

The test dataset we chose for Typograph consists of all English Wikipedia articles, which we downloaded as a single XML file [26]. This file is roughly 40gb, and at any given time of download includes slightly over 4 million articles (the XML dump also includes navigation, disambiguation redirects, categorization, templates, stubs, and more undesirable pages bringing the total to over 12 million potential articles within the XML). Each article page contains primarily narrative text organized into sections, but also includes formatting and metadata as wiki markup.

Figure 2 provides an overview of our methods used to process data from the raw Wikipedia articles to produce a 2-dimensional spatial layout of information. This process is described below.

1) Initial Text Processing

Our goal is to derive a clean corpus of narrative text for only the topic-bearing articles. The XML dump file is processed in parallel using the distributed computing framework, Hadoop, to extract article names and text fields for insertion into a high performance distributed key value store called Accumulo. Each raw article is evaluated to determine if it is truly a topic-bearing article or one of the undesirable page types based on the Wikipedia definition of an article leaving us with approximately 4 million articles. The text content of good articles is processed using Hadoop through a series of regular expression rules to remove wiki markup while preserving article content. The final article title and text content are inserted into the key-value store as ArticleTitle->Text mapping.

2) Entity Extraction and Weighting

The raw text extracted from the previous steps for each article is processed into a Lucene [8] index with a custom analyzer which applies the Rapid Automatic Keyword Extraction (RAKE) [9] algorithm. RAKE extracts keywords from individual documents automatically. RAKE operates by using a list of stopwords and phrase delimiting punctuation to split a document into an array of candidate keywords. A co-

occurrence graph is generated from individual tokens contained in candidate keyword. Candidate keywords are ranked by computing the ratio of degree to frequency for their constitute tokens. The top ranking candidates are kept as extracted keywords which represent the most essential keywords for characterizing each document. RAKE has several properties which make it ideal for our work. RAKE does not require training, it is high performing, it can be distributed, and it discovers context-rich keywords automatically. Another advantage of RAKE is that it provides a dataset-independent characterization of the document, which is important in applications in which the composition of the dataset is continually changing.

The Lucene index is valuable for search and retrieval and it offers an efficient platform for statistical analysis of all keywords in the corpus. These extracted keywords offer a more context-rich feature set than single whitespace delimited keywords. From the many hundreds of thousands of unique keywords extracted from the four million Wikipedia articles, 10,000 keywords are selected.

To create keyword vectors, the pair-wise association is measured among the 10,000 keywords to form a square similarity matrix. This provides us with a set of vectors representing our keywords consisting of dimensions which measure the relationship to other keywords. This 10,000 dimensional concept vector space is then used to spatialize the data. Additionally, each of the keywords are weighted based on their relevance to the dataset being analysed. The method for the generating the weighting schema is also a part of RAKE, the details of which are out of the scope of this paper and described in [9].

3) Generating the Spatial Layout

To spatialize the keyword vectors for visualization in 2 dimensions, we applied a self-organizing map (SOM) [36][27] as our dimension reduction method (although other algorithms could be used in place of the SOM). The complete set of keyword vectors is run through the SOM algorithm, modified to be recursive (and for our purposes, parallelized). The SOM is initialized with a 5x5 grid of sites (or neurons). The vectors of each site are then recursively run through the SOM of size 5x5 again to further distinguish this subset of the information. The recursion terminates when not enough data items remain to recurse. The coordinates produced in each recursive site are then normalized from their local coordinate set to the global coordinate set, relative to their overall place in the top level SOM. We made the design decision to allow each SOM session to run for 100 iterations, although datasets and data sizes may require a different parameter.

The total processing time from parsing the XML file to the end of SOM creation was approximately 15 hours on our server infrastructure and hardware. However, this performance could be optimized by parallelization of the keyword extraction and clustering processes.

4) Snippet and Document Retrieval

Apache Accumulo [28] is used for article storage and retrieval. When the client requests information from the server, it is retrieved from Accumulo via its title (used for the index). In the case of an article request, the text returned from Accumulo is handed to the client as-is. In the case of a snippet request, the full article text is run through a custom Lucene document highlighter which extracts a snippet of desired word length that is most dense with the supplied keyword from the full article text.

B. User Interface and Functionality

1) Implementation

Typograph is written in Java and utilizes the Eclipse Rich Client Platform (RCP) to provide the core menu and window management system. The visualization itself was created using the Scientific Visualization Framework (SVF), which is a visualization library created at PNNL to support development for Java Bindings for OpenGL (JOGL).

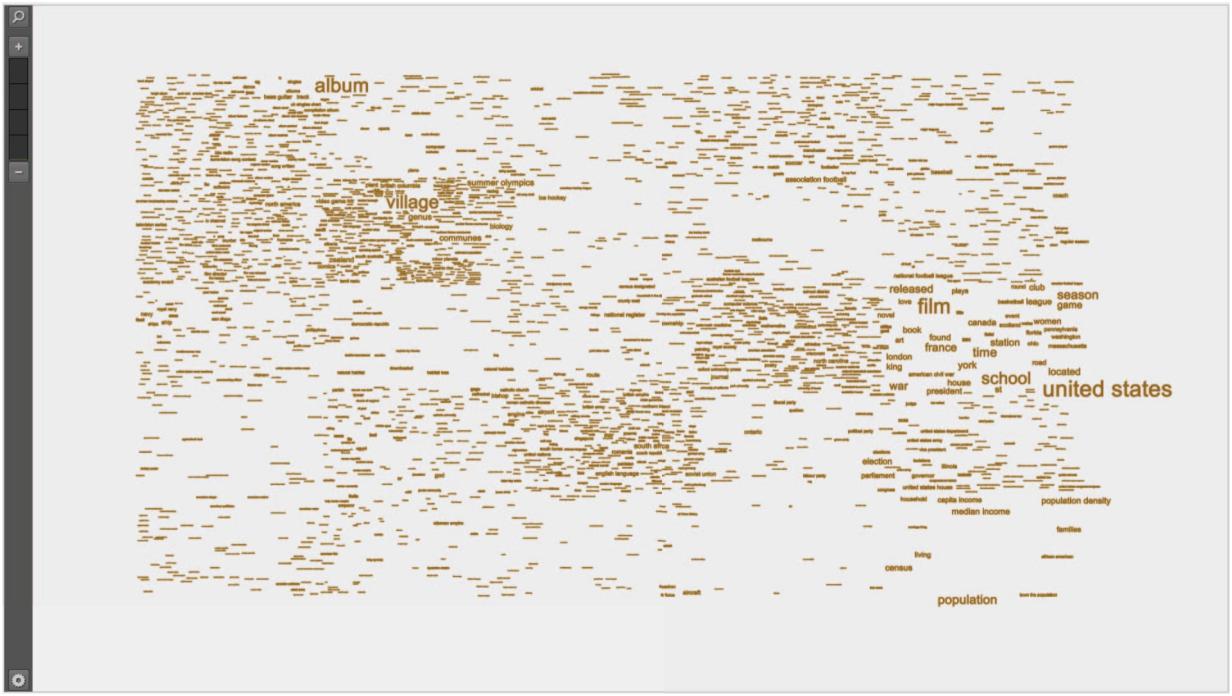

Figure 3. Screenshot of Typograph visualizing all of the English Wikipedia. Groups are clustered based on their similarity (near = similar) and sized based on their importance within the dataset.

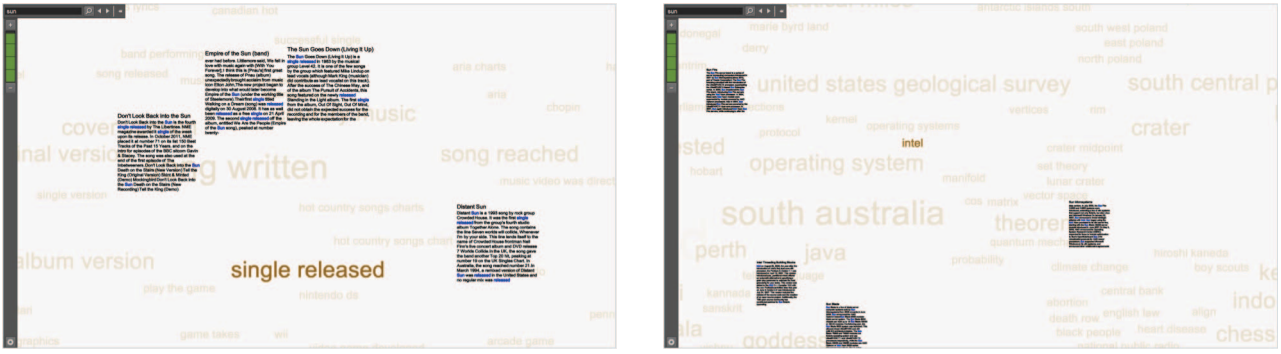

Figure 4. The snippets contextualized around each phrase or term show different uses of the search query "sun". (Left) shows how the word is used in album and song titles, while (Right) shows information about Sun Microsystems. These snippets are located within the context of the spatialization, maintaining relative similarity to the remaining information.

2) Spatialization

The Typograph user interface consists of one primary view, a spatialization of terms similar to that of a tag cloud (shown in Figure 3). The location of the information is generated via the SOM described previously. Terms and phrases are placed and rendered from the top down, starting with the highest weighted terms in each SOM site. That way, the higher weighted terms are given rendering precedence at the higher levels, and the lower, more detailed terms within the site are accessible through zooming into that location. At the highest level, terms presented are descriptors or concepts of terms at the next level in the hierarchy. The visualization traverses the hierarchy through levels of zoom until you reach the lowest level (i.e., the document level). As a result, details of higher-level terms are revealed through terms, phrases, snippets, and finally documents that describe the concept in increasing amounts of detail. The terms and phrases are rendered so that the center of the space taken up by the term or phrase is at the 2-dimensional coordinate produced by the SOM.

Terms and phrases are sized based on their weight. To reduce the visual artifacts created by longer words occupying more space because they include more characters, we visually encode the weight of a term (or phrase) by the amount of ink used to render it. This approach is based previous work called "FatFonts" [31]. Their work showed that numerical values can be encoded into the number representing the value by the amount of ink needed to render the number, providing users with the ability perceive relative value differences between displayed numbers based on a user's visual acuity. Typograph takes a similar approach, in that the weight depicts how much ink (i.e., pixels) is used to render the term or phrase. As a result, words that are short occupy more space by being bolder.

Snippets and documents are shown as more detail is requested by the user through progressively zooming into a region, as shown in Figure 6. Users can click (or touch) a term or phrase to retrieve snippets that are associated with it (described in more detail in the previous section). The top 4 snippets are placed within the spatialization according to their relative proximity to related terms and phrases. Within the snippets, the term or phrase that was clicked is highlighted (in blue) to give context to the user. Full details of the text are available to the user by displaying a document reader on demand. When reading a snippet, users can click the snippet to show the full document.

3) User Interaction

At the left of the prototype is an application bar, which contains a search button and a zoom indicator. The zoom indicator indicates the level of zoom currently shown and gives users the ability to slide or jump down more than one level at time, similar to the way zoom indicators work in online map applications. Aside from this sidebar, the content will also be the primary method for interaction (and gaining context) in an effort to minimize the interaction junk [32], and optimize the fluidity [30] of the exploration.

Users can navigate the information through multi-touch gestures traditional mouse interactions. Similar to map interfaces, zooming in and out is performed by the mouse wheel, touch pinch and zoom, or by pressing the +/- keys. Panning is performed by grabbing and sliding the canvas with the cursor, touching and dragging it, or by pressing the arrow keys in the specific direction. Typograph animates the transitions to the next level to afford the user additional context into new content as it appears. Terms at the previous level of zoom begin to fade out as the more detailed terms take their place (Figure 6). These more general terms continue to remain visible to provide context, and ultimately fade out completely as users continue to zoom in further.

Typograph supports standard text search through a search field found in the upper left corner of the prototype. Clicking the search icon will expand the search field and allow users to enter their query. Terms that do not match the search fade back revealing the search results. Search results are either exact matches or sematic matches. The exact matches are terms in the spatializations that match directly (or contain) the string in the search query. The semantic matches are terms that were extracted from documents that contain the search query. As a result, searching can provide users with an

Figure 5. Search results for the query on "sun", showing different contexts that the term appears in. Search results that are off-screen are indicated by the green bubbles on the edges, showing the direction of other search results.

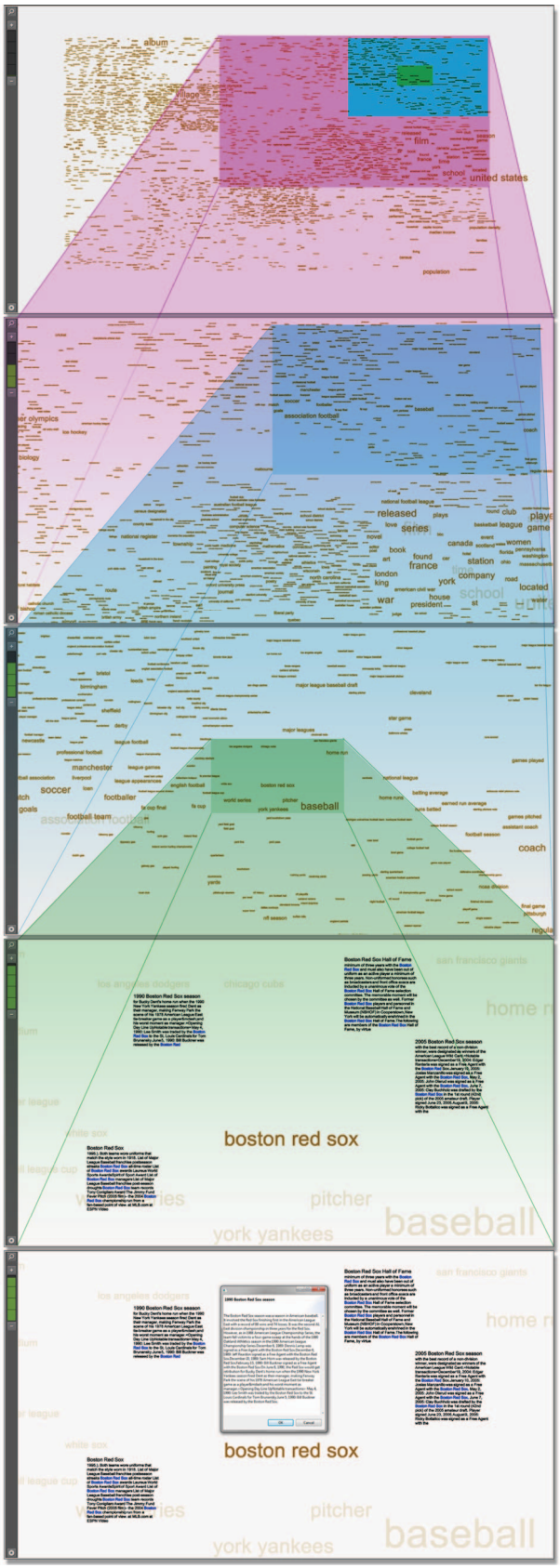

Figure 6. Progressively zooming into an area of the spatialization reveals more detailed information about that concept.

overview of the different concepts in which the search query occur. Search results that occur outside of the current zoomed in section of the spatialization are shown on the edges as green indicators (Figure 5). The user can click those indicators, which will automatically pan to the search result, or they can pan and zoom manually. Larger green indicators show terms that are a concept level higher in the zoom level while smaller indicators show terms at lower levels.

C. Use Case

We demonstrate the functionality and analytic capabilities provided by Typograph through the following use case. Consider a user interested in the different contexts in which the work "sun" is used in the English Wikipedia. Based on her understanding and domain expertise, she hypothesizes that "sun" might show up in areas including the solar system, tropical vacations, technology, etc.

She starts her exploration from the highest-level overview of terms in the dataset. From this overview, she observes the general, high-level concepts within the dataset, including: music, sports, governments, etc. In exploring these areas spatiality, she gains an understanding of how the specific regions of the spatialization correspond to different concepts. She proceeds to perform a search on the term "Sun" to reveal how the term relates across these contexts. The search results highlight phrases within different contexts (as shown in Figure 5), while making the non-results transparent to maintain contextual awareness of the regions. As she zooms and pans to terms of interest, the other matches are shown on the sides of the visualization as green indicators (Figure 5).

She finds a phrase from her "sun" query titled "single released" (Figure 4). This phrase interests her, and she chooses to investigate it further. Tapping and holding on the term reveals song lyrics that also contain the word "sun". She takes note of this use of the term "sun", and continues her search by panning to the right, towards the green search result indicators to find other contexts. She finds another search result around "Intel". Upon reading the snippets, and the terms surrounding the snippets, she gathers that these set of results focus on "Sun Microsystems". She could continue to explore other search results at this point to find additional contexts, but is satisfied with these, concluding her exploration.

IV. Discussion

A. Overview and Exploration of Large Datasets

Typograph serves as a platform for investigation of exploratory analysis and discovery in very large data collections. In extremely large data collections, it is common practice to visualize only a subset of the data content, which may be obtained by searching and filtering the dataset. In Typograph, we instead implement the information visualization mantra of "overview, zoom & filter, details on demand" [33] by providing an overview of the entire collection. This overview consists of the terms algorithmically measured to be most representative within the data collection.

When visualizing a collection of a few million pages, such as the Wikipedia example described above, this overview contains a set of terms that characterize the whole of the collection. This visualization illustrates the disproportionate volume of information concentrated in areas like popular culture and sports, as contrasted with history and science. While this provides an interesting initial perspective on the data and informs further exploration of the dataset, it is a very high-level characterization of the data collection and does not particularly serve to stimulate new questions or discoveries. The overview provides a frame in which further exploratory analysis can occur, but its general nature may limit its effectiveness in building a meaningful context in which the

data can be explored. It is an open research question as to whether the information visualization mantra should be modified when applied to very large data collections.

In addition, because the representative terms are derived without human intervention and without regard to the user's context, they are fully representative of the data but may not match well with the user's mental model of the data content. That is, a statistical summary of the information may not provide new information for a domain expert of that dataset. Instead, details within these datasets may provide the insights sought after. Thus, a mixture of domain knowledge and statistical inferences from the data may be necessary as datasets continue to increase. For example, such mixed models can be achieved by model steering or semantic interaction methods [15].

B. *Data Coverage*

A goal of visualization is to provide visual representations of structure within datasets [34]. As such, these views (especially at the overview level) are expected to provide an adequate amount of coverage. For example, for a dataset that contains 100 data objects that can be described using 200 dimensions, leveraging dimension reduction techniques that take into account each of these dimensions is a traditional approach to visualizing this information. However, as data scales continue to increase in both size and complexity, achieving coverage can be done through other means.

When combining multiple methods for scaling down sizes and complexities of datasets, the traditional concept of coverage may differ. For example, instead of using all unique words within a text dataset, entity extraction techniques may reduce the complexity prior to leveraging a dimension reduction technique. Further, extracting phrases and other patterns of unique text terms may increase the dimensionality and complexity of these datasets. What impact do such approaches have on coverage, and more importantly validity and trust of the resulting views?

The Wikipedia dataset used for describing Typograph is a good example of how the concept of coverage may change with scale. The 10,000 keywords extracted by RAKE takes the entire 4 million articles into account. However, the dimension reduction steps are then based on only a subset of the total number of unique keywords (or dimensions) in the original data. That is, 14% of the original articles cannot be characterized by any one of these extracted keywords. While the entity extraction methods could be modified to ensure better (and even complete) coverage of the dataset, the downside would be that additional, less relevant keywords would be chosen. These trade-offs should be carefully considered when designing visualizations that deal with large, complex data. Exploring such topics will continue to increase our ability to design visualizations that foster sensemaking of large and complex datasets.

V. Conclusion

In this paper, we introduce Typograph, a spatial text exploration visualization. Typograph is designed to enable users to analyze and visualize large, text datasets. The approach is based on exploring textual datasets through a term-centric spatialization, with multiple levels of detail integrated into the single view. Typograph organizes terms, phrases, snippets, and full documents based on their relative similarity to each other. As a result, clusters of information reveal themes of topics within the dataset. Typograph reveals additional information at multiple levels of detail within the spatial metaphor, and in context of the other information. The overview consists of mainly clusters of terms, and more detail is revealed as users zoom into regions of the space. As a result, the additional levels of detail are revealed in context of the terms, when demanded by the user. The multi-scale approach of Typograph is supported through a thin-client architecture, where significant data processing is performed on the server through a collection of entity extraction, information retrieval, and dimension reduction techniques. The resulting technology is one that is capable of creating insightful visualizations from large text datasets.

Acknowledgments

This work was supported by the Department of Defense. PNNL is managed for the US Department of Energy by Battelle under Contract DE-AC05-76RL01830.

References

[1] J. J. Thomas and K. A. Cook. (2005). *Illuminating the path*. Available: http://nvac.pnl.gov/agenda.stm - book

[2] P. Pirolli and S. Card, "Sensemaking Processes of Intelligence Analysts and Possible Leverage Points as Identified Though Cognitive Task Analysis " *Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, Virginia,* p. 6, 2005.

[3] C. Andrews, A. Endert, and C. North, "Space to Think: Large, High-Resolution Displays for Sensemaking," ACM CHI, 2010.

[4] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow, "Visualizing the non-visual: spatial analysis and interaction with information for text documents," presented at the Readings in information visualization: using vision to think, 1999.

[5] A. Skupin, "A Cartographic Approach to Visualizing Conference Abstracts," *IEEE Computer Graphics and Applications,* vol. 22, pp. 50-58, 2002.

[6] A. Endert, S. Fox, D. Maiti, S. C. Leman, and C. North, "The Semantics of Clustering: Analysis of User-Generated Spatializations of Text Documents," presented at the AVI, 2012.

[7] Y.-a. Kang and J. Stasko, "Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study," in *Visual Analytics Science and Technology (VAST)*, Providence, RI, 2011.

[8] http://lucene.apache.org/core/. 2013.

[9] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents," presented at the Text Mining, 2010.

[10] J. Wang, "Clustered Layout Word Cloud for User Generated Online Reviews," Virginia Polytechnic Institute and State University, 2012.

[11] (2008, October 1). *Alias-i. 2008. LingPipe 4.0.1.* Available: http://alias-i.com/lingpipe

[12] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: an Architecture for Development of Robust HLT Applications," presented at the Annual Meeting of the Association for Computational Linguistics (ACL), 2002.

[13] A. Endert, C. Han, D. Maiti, L. House, S. C. Leman, and C. North, "Observation-level Interaction with Statistical Models for Visual Analytics," presented at the IEEE VAST, 2011.

[14] S. M. Drucker, D. Fisher, and S. Basu, "Helping users sort faster with adaptive machine learning recommendations," presented at the Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III, Lisbon, Portugal, 2011.

[15] A. Endert, P. Fiaux, and C. North, "Semantic Interaction for Visual Text Analytics," ACM CHI, 2012.

[16] J. S. Risch, D. B. Rex, S. T. Dowson, T. B. Walters, R. A. May, and B. D. Moon, "The STARLIGHT information visualization system," presented at the Proceedings of the IEEE Conference on Information Visualisation, 1997.

[17] J. Alsakran, Y. Chen, Y. Zhao, J. Yang, and D. Luo, "STREAMIT: Dynamic visualization and interactive exploration of text streams," presented at the IEEE Pacific Visualization Symposium, 2011.

[18] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams, "Visualization of a document collection: the vibe system," *Inf. Process. Manage.,* vol. 29, pp. 69-81, 1993.

[19] O. Kaser and D. Lemire, "Tag-cloud drawing: Algorithms for cloud visualization," *Proceedings of the World Wide Web Workshop on Tagging and Metadata for Social Information Organization,* 2007.

[20] C. Seifert, B. Kump, W. Kienreich, G. Granitzer, and M. Granitzer, "On the Beauty and Usability of Tag Clouds," in *Information Visualisation, 2008. IV '08. 12th International Conference*, 2008, pp. 17-25.

[21] J. Feinberg, *Wordle*. http://www.wordle.net, 2009.

[22] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and Q. Huamin, "Context preserving dynamic word cloud visualization," in *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, 2010, pp. 121-128.

[23] F. V. Paulovich, F. M. B. Toledo, G. P. Telles, R. Minghim, and L. G. Nonato, "Semantic Wordification of Document Collections," *Comp. Graph. Forum,* vol. 31, pp. 1145-1153, 2012.

[24] S. J. Attfield, S. K. Hara, and B. W. Wong, "Sensemaking in Visual Analytics: Processes and Challenges," in *EuroVAST 2010: International Symposium on Visual Analytics Science and Technology*, 2010, pp. 1-6.

[25] M. Pohl, M. Smuc, and E. Mayr, "The User Puzzle: Explaining the Interaction with Visual Analytics Systems," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 18, pp. 2908-2916, 2012.

[26] http://en.wikipedia.org/wiki/Wikipedia:Database_download 2013.

[27] J. Salonen, "Self-organising map based tag clouds," presented at the Proceedings of the 1st OPAALS Conference, November 26-27, 2007, Rome, Italy, 2011.

[28] http://accumulo.apache.org/. 2013.

[31] M. Nacenta, U. Hinrichs, and S. Carpendale, "FatFonts: combining the symbolic and visual aspects of numbers," presented at the Proceedings of the International Working Conference on Advanced Visual Interfaces, Capri Island, Italy, 2012.

[32] A. Endert and C. North, "Interaction Junk: User Interaction-Based Evaluation of Visual Analytic Systems," BELIV: Beyond Time and Errors - Novel Evaluation Methods for Visualization, 2012.

[33] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, 1996, pp. 336-343.

[34] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds., *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999, p.^pp. Pages.

[35] Nocaj, A.; Brandes, U. Organizing search results with a reference map. IEEE Transactions on Visualization and Computer Graphics, v. 18, p. 2546-2555, 2012

[36] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.