HyperMap: a visual analytics approach for analyzing large collections of unstructured text

Sam Yu-Te Lee*
University of California, Davis

Aryaman Bahukhandi†
University of California, Davis

Kwan-Liu Ma‡
University of California, Davis

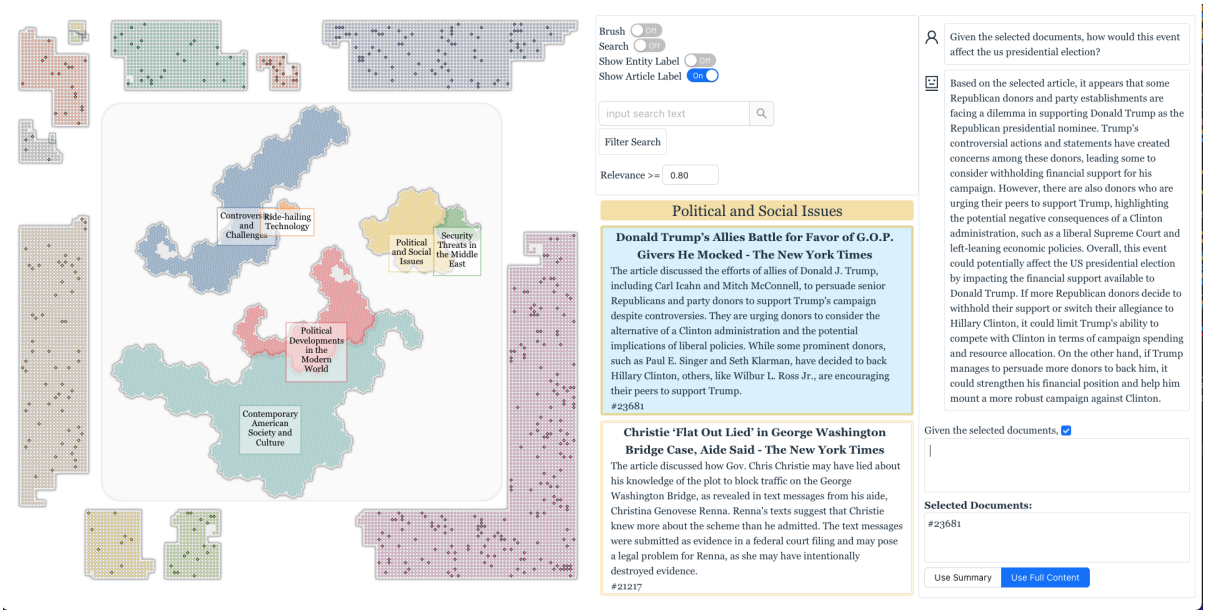


Figure 1: The hypergraph visualization generated by combining generalized Hilbert curves and Gosper curves.

Abstract

Sensemaking on large collections of unstructured text (corpus) is a challenging task that analysts have to perform. Previous works approach this problem either from a topic- or entity-based perspective, but they lack interpretability and trust. In this paper, we propose a visual analytics approach that combines topic- and entity-based techniques seamlessly by modeling the corpus as a hypergraph. The hypergraph is then hierarchically clustered with an agglomerative clustering algorithm by combining semantic and connectivity similarity. We visualize the clustering result to allow analysts to explore and reorganize a corpus for their analysis. The system is designed to foster interpretability and trust by providing rich semantic context for the visualization and by supporting curating interactions. Case studies and a task-based evaluation are used to demonstrate the effectiveness and trustworthiness of the system.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

Text data is ubiquitous in the modern world. From news articles and social media posts to scientific publications, the tremendous amount of text data that is produced poses a great challenge to anyone who needs to analyze them. Visual analytics (VA) mitigates this challenge by combining mathematical models and visualizations to automate the process and reduce the cognitive load. *Model Alignment*, proposed by Chuang et al. [10], refers to the alignment of analysis tasks, visual encodings and model decisions. Failure to align them impairs users' interpretation and trust in visual analytic systems. However, in text analysis, the available models often align poorly with analysis tasks. For example, topic models are commonly used to model the topical structure of text documents, where a *topic* is characterized as a probabilistic distribution spanning a given vocabulary [47]. This transformation from *topics*, a high-level concept that the user seeks to understand, to a *probabilistic distribution*, a low-level concept that mathematical models can operate on, prevents proper model alignment. The misalignment between analysis tasks and models limits the usage of visual analytics systems for users who are not familiar with the underlying models.

Recent advances in large language models (LLM) present a promising solution to this problem. LLMs have proven successful in various natural language processing (NLP) tasks, including Information Extraction (IE). Li et al. [27] evaluated ChatGpt's capabilities on IE tasks comprehensively, and found that it excels under an OpenIE setting, where the model relies solely on user input to extract information from documents. The capability of LLMs to extract information from documents according to user intent eliminates the need to carefully align the analysis tasks and models in VA systems. In the previous example, instead of relying on abstruse and unfathomable probabilistic models, LLMs can directly process the text data and summarize the topics of the documents. A user can directly ask a LLM: *'What are the topics of these articles?'*, and the LLM would give a human-like response, such as *'The articles are about . . . '*.

*e-mail: ytlee@ucdavis.edu
†e-mail: abahukhandi@ucdavis.edu
‡e-mail: klma@ucdavis.edu

Following the above discussion, we propose a novel VA system that allows users to explore, reorganize and analyze large collections of unstructured text data. The system is built upon an LLM-based information extraction pipeline, which is capable of extracting topics and salient entities (or concepts) from a given corpus. We demonstrate how we carefully design the pipeline to address certain limitations of LLMs. The result is then modeled as a hypergraph, hierarchically clustered, and visualized as an interactive bipartite graph using space-filling curve layouts with rich interaction supporting expansion, deletion, and searching. Finally, users can directly use the reorganized corpus to query LLM for detailed document analysis.

The contributions of this paper are as follows:

- We propose an LLM-based information extraction pipeline that is capable of extracting topics and salient entities from a given corpus.
- We propose a novel bipartite space-filling curve layout that is capable of visualizing clusters in large hypergraphs.
- We propose a novel VA system that allows users to explore, reorganize and analyze large collections of unstructured text data.

2 Related Works

2.1 Visual Analytics of large collections of text

Topic-based approaches Topic-based approaches employ certain variations of topic models to organize the documents in a meaningful way. Each topic is often presented as a 'bag of words', which can be in the form of a sequence of words [2, 8, 16, 19, 25, 50, 51] or word clouds [8, 33]. The modeling result provides an overview of the dataset for subsequent analysis tasks. Despite their variations in model choices, the use of topic models as an overview, as well as its 'bag-of-word' visualization, is reported by Lee et al. [26] to be problematic in a comprehensive user study, especially for non-expert users. First, the discovered topics are not always semantically meaningful in human evaluation [7]. Second, the 'bag of words' representation causes misinterpretation of the modeling results. Users sometimes overlooked important words, assumed association between adjacent words, or misinterpreted the meaning of a word due to lack of context. Chuang et al. [10] concluded that these problems arise from a misalignment between the analysis task, visual encoding and model. Most systems employ topic models due to a need to provide an overview of the documents. When topic models are employed to conduct such an analysis task, a transformation from semantically meaningful *topics* to probabilistic distribution over words is done. Most systems design the visualization in a way that this transformation is hidden for non-expert users because it is unnecessarily complex and unrelated to the analysis task. However, the sensemaking process becomes challenging without a basic understanding of the model because the 'bag-of-words' representation is too far away from the user's mental picture of a topic. This misalignment limits the usage of topic models for non-expert users and makes the system prone to produce false positives.

Entity-based approaches A line of work that makes successful model alignments is the entity-based approach. 'Entities' usually include named entities (people, organizations, locations), or meaningful concepts known to an existing knowledge base. The earliest of such approaches is Jigsaw [44], where entities are linked if they appear in the same document. FacetAtlas [6] generalizes the idea of entity to 'facets' which can be entities or any keywords or user's interest. ConceptVector [36] uses 'concept' to represent a similar idea. Although the system is also built for document analysis, it focuses on enabling users to build these concepts in a semi-automatic manner rather than extracting them fully automatically. Generally, entity-based approaches exhibit better model alignments than topic-based approaches [10], but the polysemy of natural language makes them prone to produce false positives [36].

Embedding-based approaches Finally, an important line of work organizes documents by directly modeling their semantic similarity [45]. Documents are first projected into a high-dimensional vector space where similarity can be measured, and then a dimensionality reduction technique (e.g.t-SNE) is used to project the dataset onto a two-dimensional space for visualization. Proximity in the reduced dimension represents similarity between documents. Earlier works construct a sparse vector using term-frequency based scores such as *TF-IDF* or BM25 [9, 42]. More recently, the success of pre-trained language models like BERT [14] popularizes the idea of embedding documents in a dense vector space [31, 38, 46]. The embedding can then be used for document retrieval [20, 22] or visualization. Embedding-based approaches also exhibit healthy model alignment, as the vector space directly models the analysis task (finding similar documents). However, the result often lacks explainability and prevents users from trusting the result. (TODO: cite here to support the claim?)

2.2 LLMs for Information Extraction

Information Extraction aims to identify structured information of interest from unstructured text data. Some of its subtasks include Named Entity Recognition (NER), Relation Extraction (RE) and Event Extraction (EE) [32,49]. Although LLMs have proven successful in many NLP tasks, their application to IE is non-trivial. First, the *faithfulness* of LLMs needs to be carefully evaluated. Faithfulness refers to the ability of a model to adhere to the provided information and not use parametric knowledge learned during training to answer user questions [53]. When conducting information extraction, it is necessary to ensure that the extracted information is actually from the provided text and not from the model's parametric knowledge. Second is the *hallucination* problem of LLMs, where LLMs provide answers factually contradicting to input text (intrinsic) or even factually false (extrinsic). In the context of IE, we mainly focus on the intrinsic hallucination problem. A recent evaluation conducted by Bang et al. [4] found that ChatGPT rarely exhibits intrinsic hallucinations, including the abstractive summarization task from which neural models usually suffer.

More specifically, Li et al. [27] comprehensively evaluated the capabilities of ChatGPT for common IE tasks. They found that ChatGPT excels under the Open-IE setting, where the model relies solely on user input to extract information from documents, but performs poorly under the Standard-IE setting, where ChatGPT is instructed to choose a correct label. Their findings agree with Zhang et al. [52] where ChatGPT is reported to perform poorly on extractive summarization. A common reason for the poor performance of ChatGPT in these tasks is that they are essentially supervised learning tasks, and ChatGPT is not trained to perform them. To make the best use of ChatGPT (or more generally, LLMs) for IE tasks, we need to carefully design the extraction tasks as question-answering tasks instead of supervised learning tasks.

2.3 Hypergraph Visualization

A hypergraph is a generalization of a graph in which an edge can connect any number of nodes. Fischer et al. [17] divides existing static hypergraph visualization approaches into node-link-based and matrix-based approaches. Although matrix-based approaches are known for their visual scalability, they are not suitable for our system because of their lack of support for visualizing hierarchical clusters and user interactions. We thus focus on node-link-based approaches in this section.

Node-link-based approaches are the most common and intuitive way of visualizing hypergraphs, as they directly extend existing node-link visualization on graphs to hypergraphs. One line of work treats hyperedges as set membership relations between nodes, thus visualization of set memberships can be directly employed. Many works visualize set memberships by extending the Euler diagram [12, 41,43] using colored contours to indicate different sets, but the visual scalability is limited to small hypergraphs. KelpDiagram [15] and KepFusion [29] use a kelp metaphor and improve visual scalability by optimizing link sizes. However, they do not consider optimizing point positions and thus are most suitable for geospatial applications. MetroSets [21] uses a metro map metaphor in which hyperedges are represented as metro lines and nodes are represented as stations. By optimizing the point positions to mimic a metro map, they achieved high visual scalability and pleasing aesthetics. Kerren et al. [23] uses a radial layout where points are arranged in a circle arcs around the circle represent hyperedges. However, the quadratic runtime limits the system to no more than 200 vertices and 20 hyperedges.

A second line of work represents hyperedge as a polygon so the vertices of the hyperedges are also the vertices of the polygon [34,39, 40]. Colors of the polygon are used to indicate the cardinality of the hyperedge. However, these approaches have a serious overlapping issue on non-planer hypergraphs, thus limiting their scalability.

Finally, some node-link-based approaches visualize hypergraphs by converting them into graphs. The most naive way of converting is through clique expansion, where each hyperedge is expanded into a clique. This conversion is known to introduce dense edge crossings and ambiguity, as one can not tell whether two nodes are connected by a hyperedge or by a path of hyperedges [35]. Instead, Ouvrard et al. [35] proposed an extra-node representation, where extra nodes are added to represent hyperedges, essentially transforming the hypergraph into a bipartite graph. We find the extra-node representation most scalable and intuitive, thus our design direction follows this approach.

3 Design Rationale

What DRs are needed for analysts to explore and reorganize a corpus for their analytical tasks and why?

3.1 Design Considerations

- **DC1: An overview of the topic structure**
- **DC2: Support for curation through user interaction**
- **DC3: Detailed analysis of the curated result**

4 Methodology

4.1 Modeling

4.1.1 Hypergraph Construction

A hypergraph is a generalization of a graph in which an edge can connect any number of nodes [17]. A hyperedge thus represents a multi-way relationship between nodes. In this paper, we model two types of hypergraphs: article hypergraph and participant hypergraph, where articles and participants are the nodes, respectively. *Participants* are the core components that the article's content discuss [**?**]. For example, in a news article, the participants can be named entities such as people, organizations, or locations. In a research article, the participants can be the concepts or techniques used in the article.

Conducting analysis on article hypergraph and participant hypergraph correspond to topic-based and entity-based analysis, respectively. Following the definition of a hypergraph node, a hyperedge can be used to represent two types of multi-way relationships: (1) A hyperedge between *articles* can be constructed if the articles all mention the same participant. In this case, the hyperedge represents the co-mention of a participant, i.e. a named entity or a concept; (2) A hyperedge between *participants* can be constructed if the participants are mentioned together in the same article. In this case, the hyperedge represents a co-occurrence relationship between participants. Once the two hypergraphs are constructed, they are hierarchically clustered separately. Clusters in the article hypergraph represent topics that are discussed in the dataset. Clusters in the participant hypergraph represent participants (entities or concepts) that frequently co-occurred in an article. For better interpretability of the clustering result, we further assign *tags* for each cluster, which is further explained in Sect. 4.2.4.

Although these two types of hyperedges are constructed differently, we utilize the *dual* of a hypergraph to simplify the construction process. The dual of a hypergraph is simply another hypergraph, where the hyperedges are now nodes and the nodes are now hyperedges. (Add formulas here to explain). Therefore, we first model the articles as nodes and participants as hyperedges to construct the article hypergraph H_A. The participant hypergraph H_P can then be easily constructed by taking the dual of H_A. This construction process also allows us to use the same clustering algorithm on both hypergraphs, which is further explained in Sect. 4.1.2.

4.1.2 Hierarchical Clustering

Common clustering algorithms for graphs consider only graph connectivity. However, for the best interpretability of the clustering result, the node embeddings must be also used in the clustering process. The necessity of incorporating node embeddings is further explained in Sect. 4.2.4. Therefore, this limits our choice of clustering algorithms to attributed node clustering algorithms.

Although there are existing approaches that can cluster attributed nodes on graphs such as EVA [11] and iLouvain [13], they are not designed for hypergraphs. In general, hypergraphs can be clustered in two different ways: (1) Directly operate on the hyperedges by generalizing the graph clustering algorithms. For example, Kamiński et al. [5] generalizes the modularity metric for graphs to hypergraphs; (2) First transform the hypergraph into a graph and then apply normal graph clustering algorithms [24]. Although the first approach is more intuitive, it is less scalable and hard to incorporate node attributes. Thus, we decided to design our clustering algorithm following the second approach.

Considering all the above, we implemented our hierarchical clustering algorithm by first transforming the hypergraph into a graph following the edge re-weighting process proposed by Kumar et al. [24], then an agglomerative clustering algorithm [45] is applied on the re-weighted graph. In agglomerative clustering, the key is to define the similarity between nodes and similarity between clusters. We can easily incorporate node attributes into the clustering process by defining the similarity between nodes and clusters as a combination of attribute similarity S_s and connectivity similarity S_c. Since we're dealing with texts, we refer to the attribute similarity between nodes as semantic similarity.

The semantic similarity $S_s(i,j)$ is the cosine similarity of the embeddings of the two nodes, denoted as v_i. For article nodes, the embeddings are generated using the article content. For participant nodes, the embeddings are generated using a description note of the participant. More details about the embeddings are explained in Sect. 4.2.2. The connectivity similarity S_c is the weighted Topological Overlap (wTO) [18], which is a weighted generalization of the Overlap Coefficient [48], as shown in Equation 1.

$$S_s(i,j) = \frac{v_i \cdot v_j}{||v_i|| \cdot ||v_j||}, \quad S_c(i,j) = \frac{\sum_{u=1}^{N} w_{i,u} w_{u_j} + w_{i,j}}{\min(k_i, k_j) + 1 - |w_{i,j}|} \tag{1}$$

where $k_i = \sum_{j=1}^{N} |w_{i,j}|$ is the total weight of the edges connected to node i. Finally, a weighting factor α is used to balance the two similarities, as shown in Equation 2.

$$S = \alpha S_s + (1-\alpha) S_c \tag{2}$$

For the similarity between clusters, we used centroid similarity, i.e. the similarity between two clusters is the similarity between the centroids of the two clusters. The algorithm is presented in (TODO: add algorithm pseudocode here)

4.2 Preprocessing

The Methodology can work for any unstructured dataset

4.2.1 Summarization

Chatgpt for summarization

4.2.2 Document Embedding

OpenAI's embedding API

4.2.3 Participant Extraction

Chatgpt for major participant extraction and another model for entity linking

4.2.4 Topic Assignment

Chatgpt to assign topics to each cluster

5 Visualization

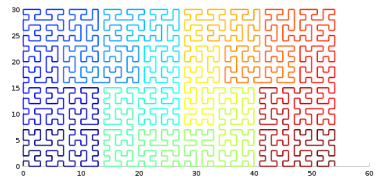
5.1 SFC for HyperGraph

In the design guidelines proposed by Abdelaal et al. [1] in a recent network visualization evaluation study, node-link-based approaches are recommended when: (1) tasks involve the identification of network clusters, and (2) the network is sparse. Condition (1) is fulfilled as explained in Sect. 3, and (2) is guaranteed by the main participant extraction process described in Sect. 4.2.3. Therefore, we decided to use node-link-based approaches for our system.

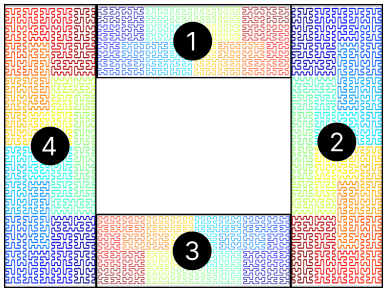
Although there are a variety of node-link-based approaches for hypergraph visualization, we find the extra-node representation proposed by Ouvrard et al. [35] most flexible and intuitive. An extra-node representation improves existing clique-expansion of hypergraphs by adding extra nodes to represent hyperedges. The extra-node representation effectively transforms the hypergraph visualization problem into a bipartite graph visualization problem. After that, any node-link-based graph visualization method can be applied. In our system, we use the space-filling curve (SFC) layout method to layout the extra-node representation of the hypergraph. The SFC layout method uses pre-computed clustering to order nodes in a sequence and then applies a space-filling curve on the node sequence to map it to a two-dimensional screen space [30]. SFC approaches are known for their efficiency and aesthetics in visualizing large graphs [28]. After the preprocessing and modeling stage described in Sect. 4, we have two hypergraphs: the article hypergraph H_A and the participant hypergraph H_P, each having its hierarchical cluster. Combining the extra-node representation and SFC layout, we visualize the article hypergraph H_A and the participant hypergraph H_P as two separate SFCs, as shown in Fig. 1.

Specifically, we divide the layout space into two parts: the peripheral and the center area. For the peripheral area, we concatenate four generalized Hilbert (Gilbert) curves [54]. A Gilbert curve is a generalized version of the Hilbert curve that can traverse any rectangular region in a way similar to the Hilbert Curve. In Fig. 2a, the Gilbert curve starts from the lower left (dark blue) and ends at the lower right (dark red). Through rotation and flipping, the start and end curve points for neighboring Gilbert curves can be concatenated smoothly, as shown in Fig. 2b. The use of concatenated Gilbert curves allows us to fill the peripheral space while having the efficiency and aesthetics of SFC layouts.

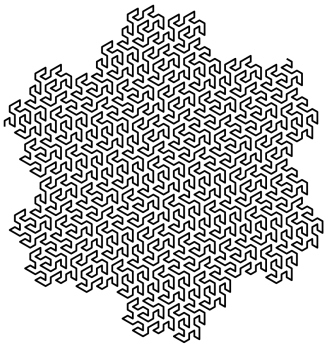
The curve to be used for the center area is technically unbounded. In early prototyping, we found that using the same curve as the peripheral region was confusing for the user, as it was hard to distinguish between the peripheral and center areas. We decided to use a



(a) One Gilbert curve



(b) Four concatenated Gilbert curves



(c) An order-4 Gosper curve

Figure 2: (a), (b): Illustration of concatenating Gilbert curves. The color indicates the traverse direction of Gilbert curves: start with dark blue and end with dark red. (c): An example of order-4 Gosper curve used for the center area.

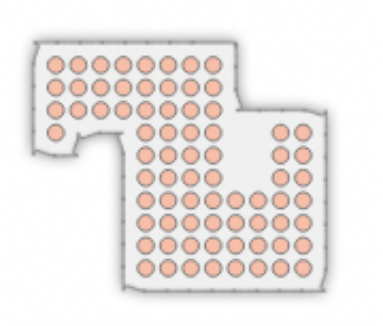
simple Gosper curve (Fig. 2c) to layout the nodes for better aesthetics. The resulting visualization looks similar to GosperMap [3], but we did not employ the advanced techniques proposed in GosperMap. The interactions to support the exploration and reorganization of the dataset are the main focus of the system, which are also not limited to any specific curve.

After the curves are generated, we can apply the curves on the node sequences to generate the two-dimensional layout. We chose to put the article hypergraph in the center area because the articles are the main analysis targets for the user. Consequently, the participant hypergraph is put in the peripheral area.

5.2 Improving the readability

Spacing Strategy We employ a simple spacing strategy.

Concave hull approximation After applying the SFC layout, we use a concave hull algorithm [37] to generate an approximation polygon for each cluster. The polygons are used to generate borders and calculate label positions for the clusters. The concave hull algorithm is generated based on a cluster of points, in our case, the nodes in a cluster. This means that the algorithm can be applied



(a)

(b)

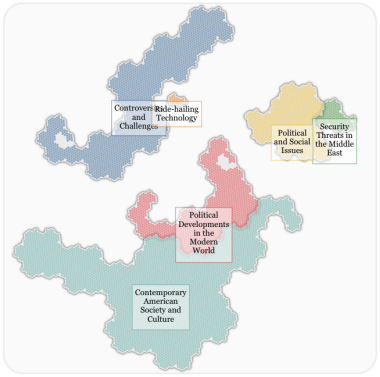
Figure 3: (a) An example of Gilbert cluster border (b) An example of Gosper cluster borders

to any curve we used for the SFC layout. This is a desirable property because we are using two different curves in our system, and the center area curve choice is flexible. Using the same algorithm guarantees a unified aesthetic across curves.

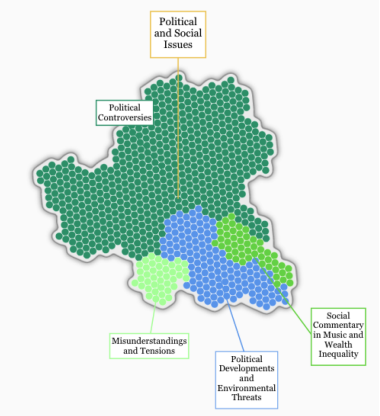
The original concave hull algorithm is designed for approximating points, but we need to approximate circles. Naively we can use the center of the circles as the points for the algorithm, but this would result in a polygon that is too small and crosses the circles on the boundary. To address this issue we use a simple trick: we add extra points to the cluster by extrapolating the original points. For a Gilbert curve, the curve moves perpendicularly, so the resulting polygon would have perpendicular corners. We can therefore add eight extra points around each original point so that the extrapolation forms a three-by-three grid. On the boundary of the cluster, these extra points prevent the concave hull from passing across the original points. Since the concave hull algorithm has an $O(n \log n)$ time complexity, the performance overhead introduced by the extra points is negligible.

Borders The borders are generated by applying a smoothing algorithm on the polygons. For Gosper curves, we use the polygon as control points to generate a cubic basis spline as the border. For Gilbert curves, we use a similar approach but with a cubic Bezier curve. More specifically, for each pair of consecutive points, we use a smoothing factor to interpolate the control point. This results in a sketchy style at the border corners. Two examples are given in Fig. 3.

Labeling Labeling the clusters is essential for users to explore the dataset. We use the topic assignment described in Sect. 4.2.4 to label the clusters. When using SFC layouts, determining the



(a) Label position of the cluster is the centroid of each cluster.



(b) Labels of the sub-clusters of a cluster. The color indicates different sub-clusters.

Figure 4: (a) Labels (topics) of the article clusters (b) An example of expanded cluster labels

label position automatically is challenging because the shape of the clusters can be irregular. For a cluster, the label position is simply the centroid of the polygon. When a cluster is expanded through user interaction, the sub-clusters within need to be clearly labeled as well. Using the centroid of the sub-cluster as the label position is not a good choice because the label would cause a serious cluttering issue. Therefore, for a sub-cluster, we first calculate the centroid of the sub-cluster, and then we extend the line from the parent cluster centroid to the sub-cluster centroid. Once the intersection point of the extended line and the parent border is found, we extend the line by a fixed amount to avoid any overlapping issues. This results in a radial layout for the sub-cluster labels, as shown in Fig. 4. The generalizability of the concave hull algorithm makes our labeling position calculation applicable to any curve we use for the SFC layout.

5.3 Design Choices

Why show all the nodes as circles

Why hide the links

6 System Design

6.1 Cluster View

Use SFC hypergraph to show topical structure of the dataset.

6.1.1 Interactions

Click

Expansion

Filtering

Searching

6.2 Article View

6.3 Analysis View

ACKNOWLEDGMENTS

The authors wish to thank A, B, and C. This work was supported in part by a grant from XYZ.

REFERENCES

[1] M. Abdelaal, N. D. Schiele, K. Angerbauer, K. Kurzhals, M. Sedlmair, and D. Weiskopf. Comparative evaluation of bipartite, node-link, and matrix-based network representations. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):896–906, 2022.

[2] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher. Serendip: Topic model-driven visual exploration of text corpora. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 173–182, 2014. doi: 10.1109/VAST.2014.7042493

[3] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, and A. Saulnier. Gospermap: Using a gosper curve for laying out hierarchical data. *IEEE transactions on visualization and computer graphics*, 19(11):1820–1832, 2013.

[4] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[5] P. P. . F. T. Bogumił Kamiński. Community detection algorithm using hypergraph modularity. In *Complex Networks & Their Applications IX: Volume 1, Proceedings of the Ninth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2020*, pp. 152–163. Springer, 2021.

[6] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE transactions on visualization and computer graphics*, 16(6):1172–1181, 2010.

[7] J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22, 2009.

[8] I. Cho, W. Dou, D. X. Wang, E. Sauda, and W. Ribarsky. Vairoma: A visual analytics system for making sense of places, times, and events in roman history. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):210–219, 2016. doi: 10.1109/TVCG.2015.2467971

[9] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013. doi: 10.1109/TVCG.2013.212

[10] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 443–452, 2012.

[11] S. Citraro and G. Rossetti. Eva: Attribute-aware network segmentation. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pp. 141–151. Springer, 2020.

[12] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE transactions on visualization and computer graphics*, 15(6):1009–1016, 2009.

[13] D. Combe, C. Largeron, M. Géry, and E. Egyed-Zsigmond. I-louvain: An attributed graph clustering method. In *Advances in Intelligent Data Analysis XIV: 14th International Symposium, IDA 2015, Saint Etienne. France, October 22-24, 2015. Proceedings 14*, pp. 181–192. Springer, 2015.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] K. Dinkla, M. J. Van Kreveld, B. Speckmann, and M. A. Westenberg. Kelp diagrams: Point set membership visualization. In *Computer Graphics Forum*, vol. 31, pp. 875–884. Wiley Online Library, 2012.

[16] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2002–2011, 2013.

[17] M. T. Fischer, A. Frings, D. A. Keim, and D. Seebacher. Towards a survey on static and dynamic hypergraph visualizations. In *2021 IEEE visualization conference (VIS)*, pp. 81–85. IEEE, 2021.

[18] D. M. Gysi, A. Voigt, T. d. M. Fragoso, E. Almaas, and K. Nowick. wto: an r package for computing weighted topological overlap and a consensus network with integrated visualization tool. *BMC bioinformatics*, 19(1):1–16, 2018.

[19] D. Han, G. Parsad, H. Kim, J. Shim, O.-S. Kwon, K. A. Son, J. Lee, I. Cho, and S. Ko. Hisva: A visual analytics system for studying history. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4344–4359, 2022. doi: 10.1109/TVCG.2021.3086414

[20] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

[21] B. Jacobsen, M. Wallinger, S. Kobourov, and M. Nöllenburg. Metrosets: Visualizing sets as metro maps. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1257–1267, 2020.

[22] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781. Association for Computational Linguistics, Online, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.550

[23] A. Kerren and I. Jusufi. A novel radial visualization approach for undirected hypergraphs. In *EuroVis (Short Papers)*, 2013.

[24] T. Kumar, S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran. A new measure of modularity in hypergraphs: Theoretical insights and implications for effective clustering. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pp. 286–297. Springer, 2020.

[25] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer graphics forum*, vol. 31, pp. 1155–1164. Wiley Online Library, 2012.

[26] T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28–42, 2017.

[27] B. Li, G. Fang, Y. Yang, Q. Wang, W. Ye, W. Zhao, and S. Zhang. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*, 2023.

[28] K.-L. Ma and C. W. Muelder. Large-scale graph visualization and analytics. *Computer*, 46(7):39–46, 2013. doi: 10.1109/MC.2013.242

[29] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer. Kelpfusion: A hybrid set visualization technique. *IEEE transactions on visualization and computer graphics*, 19(11):1846–1858, 2013.

[30] C. Muelder and K.-L. Ma. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1301–1308, 2008.

[31] A. Narechania, A. Karduni, R. Wesslen, and E. Wall. vitality: Promoting serendipitous discovery of academic literature. 2022.

[32] Z. Nasar, S. W. Jaffry, and M. K. Malik. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39, 2021.

[33] D. Oelke, H. Strobelt, C. Rohrdantz, I. Gurevych, and O. Deussen. Comparative exploration of document collections: a visual analytics

approach. In *Computer Graphics Forum*, vol. 33, pp. 201–210. Wiley Online Library, 2014.

[34] P. Oliver, E. Zhang, and Y. Zhang. Scalable hypergraph visualization. 2023.

[35] X. Ouvrard, J. L. Goff, and S. Marchand-Maillet. Networks of collaborations: Hypergraph modeling and visualisation. *CoRR*, abs/1707.00115, 2017.

[36] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist. Conceptvector: Text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):361–370, 2018. doi: 10.1109/TVCG.2017.2744478

[37] J.-S. Park and S.-J. Oh. A new concave hull algorithm and concaveness measure for n-dimensional datasets. *Journal of Information science and engineering*, 28(3):587–600, 2012.

[38] R. Qiu, Y. Tu, Y.-S. Wang, P.-Y. Yen, and H.-W. Shen. Docflow: A visual analytics system for question-based document retrieval and categorization. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[39] B. Qu, P. Kumar, E. Zhang, P. Jaiswal, L. Cooper, J. Elser, and Y. Zhang. Interactive design and visualization of n-ary relationships. In *SIGGRAPH Asia 2017 Symposium on Visualization*, pp. 1–8, 2017.

[40] B. Qu, E. Zhang, and Y. Zhang. Automatic polygon layout for primal-dual visualization of hypergraphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):633–642, 2021.

[41] N. H. Riche and T. Dwyer. Untangling euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1090–1099, 2010. doi: 10.1109/TVCG.2010.210

[42] E. Sherkat, S. Nourashrafeddin, E. E. Milios, and R. Minghim. Interactive document clustering revisited: A visual analytics approach. In *23rd International Conference on Intelligent User Interfaces*, pp. 281–292, 2018.

[43] P. Simonetto, D. Archambault, and C. Scheidegger. A simple approach for boundary improvement of euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):678–687, 2016. doi: 10.1109/TVCG.2015.2467992

[44] J. Stasko, C. Gorg, Z. Liu, and K. Singhal. Jigsaw: Supporting investigative analysis through interactive visualization. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 131–138, 2007. doi: 10.1109/VAST.2007.4389006

[45] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. 2000.

[46] Y. Tu, O. Li, J. Wang, H.-W. Shen, P. Powałko, I. Tomescu-Dubrow, K. M. Slomczynski, S. Blanas, and J. C. Jenkins. Sdrquerier: A visual querying framework for cross-national survey data recycling. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[47] I. Vayansky and S. A. Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, 2020.

[48] M. Vijaymeena and K. Kavitha. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3(2):19–28, 2016.

[49] W. Xiang and B. Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019.

[50] Y. Yan, Y. Tao, S. Jin, J. Xu, and H. Lin. An interactive visual analytics system for incremental classification based on semi-supervised topic modeling. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 148–157, 2019. doi: 10.1109/PacificVis.2019.00025

[51] Y. Yang, Q. Yao, and H. Qu. Vistopic: A visual analytics system for making sense of large document collections using hierarchical topic modeling. *Visual Informatics*, 1(1):40–47, 2017.

[52] H. Zhang, X. Liu, and J. Zhang. Extractive summarization via chatgpt for faithful summary generation. *arXiv preprint arXiv:2304.04193*, 2023.

[53] W. Zhou, S. Zhang, H. Poon, and M. Chen. Context-faithful prompting for large language models, 2023.

[54] J. Červený. https://github.com/jakubcerveny/gilbert/commits/master generalized hilbert ("gilbert") space-filling curve for rectangular domains of arbitrary (non-power of two) sizes., 2019.