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

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

2 RELATED WORKS

2.1 Hypergraph visualization

[5] We decided to use node-link-based representation because it is more familiar to users [1].

2.2 Summarizing large collections of text

Topic models, entity-based summarization (VA approaches)

2.3 Interpretability and Trust in text analysis

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 [5]. 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.

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.

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 [3] and iLouvain [4], 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. [2] generalizes the modularity metric for graphs to hypergraphs; (2) First transform the hypergraph into a graph and then apply normal graph clustering algorithms [7]. 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. [7], then an agglomerative clustering algorithm [8] 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

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

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) [6], which is a weighted generalization of the Overlap Coefficient [9], 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 Space Filling Curves

Introduce Gosper curve and generalized Hilbert curve, and how they are used for large graph layout

5.2 SFC for HyperGraph

Using the Gosper curve to layout the article graph

Concatenating four generalized Hilbert curve to layout the entity graph on the peripheral

5.3 Spacing Strategy

5.4 Border Approximation

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.